

**UNIVERSIDAD CARLOS III DE MADRID**

**Escuela Politécnica Superior - Leganés**

**INGENIERÍA DE TELECOMUNICACIÓN**



PROYECTO FIN DE CARRERA

**Implementación de un asistente para la mejora  
de interfaz gráfica de usuario. Validación de un  
código basado en el Método de Elementos  
Finitos**

AUTOR: DAVID GARCÍA SÁNCHEZ

TUTOR: LUIS EMILIO GARCÍA CASTILLO

DIRECTOR: DANIEL GARCÍA DOÑORO

Leganés, 2017



*A mis padres*  
*A mi familia*



---

# AGRADECIMIENTOS

En primer lugar gracias a mis padres, que desde el inicio me metieron en la cabeza eso de tener que estudiar una carrera y tuvieron que hacer un gran esfuerzo económico para que yo pudiera estudiar. Ellos que con sus ánimos una veces, y sus preguntas un tanto impertinentes otras, han hecho posible que hoy esté aquí defendiendo este PFC.

A mis amigos y compañeros de clase, con los que he disfrutado de los años jóvenes de mi vida haciendo que los tragos amargos pasaran más rápido. Esos con los que he compartido muchas horas de laboratorio y biblioteca, llorándonos las penas juntos y disfrutando de las escasas alegrías que estos años nos han ido dando. Ahora nos toca vivir bodas, nacimientos y buenos ratos alrededor de un roscón de reyes o de una cerveza.

A todos esos compañeros de prácticas de los que fui aprendiendo un poquito de cada uno de ellos. En especial a Javi compi, con el que más horas he compartido y con el que más me he quejado con diferencia. Javi tío, hemos llegado al final.

Por supuesto a mi mujer, Pili, que prácticamente se ha sacado otra carrera completa a mi lado. Sufriendo mi malhumor y teniéndose que aguantar cuando me tocaba hincar los codos, comprendiéndome siempre y dándome los ánimos que tantas veces me han hecho falta y sin los cuales no hubiera sido capaz de llegar hasta el final, gracias cariño.

A Dani, compañero de clase primero y director de proyecto después, que ha estado dispuesto a ayudarme en todo lo que he necesitado, tanto en la parte más técnica que es la que se presupone, como dándome los ánimos necesarios y alguna que otra colleja también necesaria. No tengo ninguna duda que sin tu ayuda no habría llegado hasta aquí.

A todos esos amigos y familiares que me habéis animado a lo largo de este tiempo, por suerte sois muchos y no os puedo enumerar.

Y para terminar, a la última en llegar. A mi hija Carla, que sin saberlo me ha aportado el último empujón que me faltaba. Quiero decirle que su padre termina siempre lo que empieza y que por eso, a pesar de todas las dificultades, es Ingeniero de Telecomunicaciones. Muchas gracias a todos.

David García Sánchez



---

# RESUMEN

Una de las líneas de trabajo del grupo de investigación con el que colaboro en la realización de este Proyecto Final de Carrera (PFC), es el diseño e implementación de un código de simulación electromagnética basado en el Método de los Elementos Finitos (FEM), ya sea para el análisis de radiación de antenas, cálculo de la sección Radar de diversos objetos o el análisis de estructuras de guías de onda. Además, para que este código de simulación sea utilizable, dentro del propio grupo de investigación, también se realiza el desarrollo de una interfaz gráfica de usuario (GUI) que facilita la generación de los ficheros de entrada del mismo.

Existen dos aspectos críticos para lograr que una herramienta software de este tipo pueda ser utilizada por investigadores. El primero de ellos es realizar un exhaustivo proceso de validación/verificación de sus resultados. Además, para que la curva de aprendizaje inherente en estas herramientas sea lo más rápida posible, es necesario que tenga una interfaz gráfica de usuario sencilla y, sobre todo, muy fácil de usar. Es, por tanto, en estos dos puntos donde se centra la realización del presente PFC.

Por un lado, este PFC presenta el desarrollo de un *"wizard"* o asistente que facilita el modelado geométrico de estructuras complejas, de tal forma que la creación de los puertos de excitación de las antenas o guías de onda pueda realizarse de forma sencilla y rápida. De este modo, se implementa una mejora en la mencionada GUI, ayudando a la usabilidad de la misma. La implementación del asistente se ha realizado utilizando el lenguaje de programación TCL-TK, que es el lenguaje usado previamente en el desarrollo de la GUI existente. En concreto, este *"wizard"* permite generar guías de onda y cables coaxiales en tres simples pasos, en los que el asistente pregunta al usuario por el tipo de excitación, la localización y los parámetros característicos de la estructura. Una vez completados los tres pasos, el asistente genera las entidades que forman el modelo y lo muestra en pantalla para que el usuario continúe con el proceso de simulación.

Además, completando de este modo el trabajo relacionado con la mejora de la GUI, se ha elaborado un manual de usuario, que sirve de tutorial e incluye una explicación detallada de cómo usar el asistente paso por paso mediante dos ejemplos: la creación de una antena de tipo bocina y la creación de una antena tipo parche.

Por otra parte, y tal y como se ha mencionado anteriormente, para que todo código sea confiable, debe pasar por un proceso de validación de resultados. De este modo, durante la realización de este PFC, se han validado los resultados dados por el código de Elementos Finitos, tanto en el análisis de problemas de radiación como en el análisis de estructuras de guías de ondas. Dichos resultados han sido comparados con resultados analíticos y con los resultados dados por códigos comerciales de simulación electromagnética para una correcta comprobación. Además, y para que puedan ser utilizadas en el futuro como batería de pruebas de los códigos que se implementen en el grupo de investigación con el que colaboro, todas las pruebas y simulaciones han sido documentadas, para en base a estos datos, obtener conclusiones sobre el tamaño de los elementos y los tiempos de simulación más eficientes en cada problema analizado.



---

# ABSTRACT

This project is carried out as part of a research line that consists on the design and the implementation of an electromagnetic simulation code based on the Finite Element Method (FEM). The code is able to analyze different kind of problem such as antenna radiation, radar cross section analysis and the simulation of waveguide structure. In order to ensure the usability of this code, the research group where I work doing this project, has another research line based on the implementation of a graphical user interface (GUI) that helps in the generation of the code input files.

There are two critical aspects when creating software tools to be used by researchers: the first one is to perform an accurate and comprehensive process of validation and verification of their results and the second one is to have an intuitive and easy-to-use GUI that makes the learning curve as fast as possible. This project is focusing on those two key aspects.

On one hand, this project presents the development of a "*wizard*" that facilitates the geometric modeling of complex structures, as for instance, the easy creation of antenna excitation ports or the generation of waveguide ports as well. In this way, the mentioned GUI has been improved making it more usable. The implementation of the wizard has been done using the programming language TCL-TK. This "*wizard*" allows the generation of waveguide ports and coaxial cables in three simple steps. During these steps, the wizard will ask the user which type of excitation, localization and characteristic parameters to set. Once these three steps are complete, the wizard generates the entities that form the model showing the result on the screen. Additionally to the improvement of the GUI, a user manual has been created in order to provide detailed information to the user about how to use the wizard. It contains two examples illustrating the creation of a horn antenna as well as patch antenna.

On the other hand, in order to make this code reliable, it needs to fulfill a process for the validation of the results. In this regard, the results provided by the finite element method code have been validated for the analysis of radiation problems and waveguide problems as well. These results have been compared to the analytical results and the results provided by commercial codes in order to proof their veracity.

Furthermore, all the benchmarks have been documented in order to have a report for future uses (the report can be used by future researcher of the group to verify their codes) reaching conclusions about the size of the elements and the most efficient simulation times for each benchmark.

---

# ÍNDICE GENERAL

|   |             |
|---|-------------|
| <b>Agradecimientos</b>                              | <b>v</b>    |
| <b>Resumen</b>                                      | <b>vii</b>  |
| <b>Abstract</b>                                     | <b>ix</b>   |
| <b>Índice general</b>                               | <b>xi</b>   |
| <b>Índice de figuras</b>                            | <b>xiii</b> |
| <b>Índice de tablas</b>                             | <b>xvii</b> |
| <b>Índice de pseudocódigos</b>                      | <b>1</b>    |
| <b>1 Introducción</b>                               | <b>3</b>    |
| 1.1 Antecedentes del proyecto . . . . .             | 3           |
| 1.2 Objetivos . . . . .                             | 5           |
| 1.3 Software utilizado . . . . .                    | 6           |
| 1.3.1 GiD . . . . .                                 | 6           |
| 1.3.2 Modulo FEM . . . . .                          | 8           |
| 1.4 Contenido del Proyecto . . . . .                | 10          |
| <b>2 Asistente para la creación de modelos</b>      | <b>13</b>   |
| 2.1 Introducción . . . . .                          | 13          |
| 2.2 Implementación del asistente . . . . .          | 14          |
| 2.2.1 Paquete tkwizard . . . . .                    | 14          |
| 2.2.2 Definición del asistente . . . . .            | 17          |
| 2.3 Definición de los Pasos del Asistente . . . . . | 23          |
| 2.3.1 Guía Rectangular: Localización . . . . .      | 24          |

|          |   |           |
|----------|---|-----------|
| 2.3.2    | Guía Rectangular: Parámetros . . . . .                            | 25        |
| 2.3.3    | Guía Rectangular: Creación . . . . .                              | 25        |
| 2.3.4    | Cable Coaxial: Localización . . . . .                             | 29        |
| 2.3.5    | Cable Coaxial: Parámetros . . . . .                               | 30        |
| 2.3.6    | Cable Coaxial: Creación . . . . .                                 | 31        |
| 2.4      | Conclusiones . . . . .  | 34        |
| <b>3</b> | <b>Verificación de Resultados</b>                                 | <b>37</b> |
| 3.1      | Introducción . . . . .  | 37        |
| 3.2      | Problemas de guías de ondas . . . . .                             | 37        |
| 3.2.1    | Prueba 1: Guía rectangular . . . . .                              | 38        |
| 3.2.2    | Prueba 2: Codo plano-H de 90° . . . . .                           | 40        |
| 3.2.3    | Prueba 3: Codo plano-E de 90° . . . . .                           | 41        |
| 3.2.4    | Prueba 4: Codo plano-H + plano-E de 90° . . . . .                 | 43        |
| 3.2.5    | Prueba 5: Cambio de ancho . . . . .                               | 46        |
| 3.2.6    | Prueba 6: Cambio del alto . . . . .                               | 47        |
| 3.2.7    | Prueba 7: Cambio en el alto y ancho . . . . .                     | 49        |
| 3.2.8    | Prueba 8: T-Mágica . . . . .                                      | 50        |
| 3.2.9    | Prueba 9: Unión en T . . . . .                                    | 54        |
| 3.2.10   | Prueba 10: Combinador de guías de ondas . . . . .                 | 56        |
| 3.2.11   | Conclusiones . . . . .  | 58        |
| 3.3      | Problemas de radiación/scattering . . . . .                       | 58        |
| 3.3.1    | Prueba 11: Análisis biestático de esfera recubierta . . . . .     | 59        |
| 3.3.2    | Prueba 12: Análisis monoestático de la NASA Almond . . . . .      | 62        |
| 3.3.3    | Prueba 13: Análisis de una bocina circular . . . . .              | 66        |
| 3.3.4    | Conclusiones . . . . .  | 67        |
| <b>4</b> | <b>Conclusiones</b>   | <b>71</b> |
| 4.1      | Conclusiones finales y posibles líneas de investigación . . . . . | 71        |
| <b>5</b> | <b>Presupuesto</b>  | <b>73</b> |
| <b>A</b> | <b>Manual de Usuario</b>  | <b>75</b> |
| A.1      | Bocina . . . . .  | 75        |
| A.1.1    | Creación de la geometría . . . . .                                | 76        |
| A.2      | Parche . . . . .  | 84        |
| A.2.1    | Creación de la geometría . . . . .                                | 84        |
|          | <b>Bibliografía</b>   | <b>93</b> |

---

# ÍNDICE DE FIGURAS

|      |   |    |
|------|---|----|
| 1.1  | Diagrama de bloques . . . . .   | 7  |
| 1.2  | Disposición típica de un problema FE-IIIEE . . . . .                          | 8  |
|      |   |    |
| 2.1  | Ventana del asistente básica. . . . .   | 21 |
| 2.2  | Ventana del asistente coloreada. . . . .                                      | 21 |
| 2.3  | Ventana del asistente final. . . . .  | 21 |
| 2.4  | Selección de estructuras. . . . .   | 23 |
| 2.5  | Localización de la estructura. . . . .  | 24 |
| 2.6  | Localización del Eje. . . . .   | 25 |
| 2.7  | Situación de la puerta. . . . .   | 25 |
| 2.8  | Parámetros de la estructura. . . . .  | 26 |
| 2.9  | Ventana de selección de estructuras . . . . .                                 | 30 |
| 2.10 | Localización de la estructura . . . . .                                       | 30 |
| 2.11 | Localización del Eje. . . . .   | 30 |
| 2.12 | Vector Longitudinal. . . . .  | 30 |
| 2.13 | Parámetros. . . . .   | 31 |
|      |   |    |
| 3.1  | Geometría de la guía rectangular utilizada en la primera prueba . . . . .     | 38 |
| 3.2  | Mallados de la guía rectangular . . . . .                                     | 39 |
| 3.3  | Convergencia del error y tiempo de cómputo para una guía rectangular. . . . . | 39 |
| 3.4  | Codo Plano H de 90°. . . . .  | 40 |
| 3.5  | Convergencia del error para un codo plano H. . . . .                          | 41 |
| 3.6  | Codo Plano-E de 90°. . . . .  | 42 |
| 3.7  | Mallados Codo plano E . . . . .   | 42 |
| 3.8  | Convergencia del error para un codo plano E. . . . .                          | 43 |
| 3.9  | Cascada de codo plano-H y plano-E de 90° con $d = 0$ . . . . .                | 43 |
| 3.10 | Convergencia del error para un codo plano EH con $d = 0$ . . . . .            | 44 |

|      |  |    |
|------|--|----|
| 3.11 | Mallados codo plano H + codo plano E . . . . .                           | 45 |
| 3.12 | Convergencia del error para un codo plano EH con $d = 0.5a$ . . . . .    | 46 |
| 3.13 | Cambio de ancho de la guía . . . . .                                     | 46 |
| 3.14 | Cambio de ancho de la guía . . . . .                                     | 47 |
| 3.15 | Convergencia del error para el cambio de ancho de la guía . . . . .      | 48 |
| 3.16 | Cambio del alto de la guía . . . . .                                     | 48 |
| 3.17 | Convergencia del error para el cambio de alto de la guía . . . . .       | 49 |
| 3.18 | Cambio en el alto y en el ancho de la guía . . . . .                     | 50 |
| 3.19 | Convergencia del error para el cambio de alto y en el ancho de la guía . | 51 |
| 3.20 | Módulo de la componente z del campo-H excitando el puerto 3. . . . .     | 51 |
| 3.21 | T Mágica . . . . .   | 53 |
| 3.22 | Convergencia del error para la T-Mágica . . . . .                        | 53 |
| 3.23 | Unión en T . . . . .   | 54 |
| 3.24 | Mallados de la unión en T. . . . .                                       | 54 |
| 3.25 | Convergencia del error para la unión en T . . . . .                      | 55 |
| 3.26 | Parte real de componente z del campo eléctrico. . . . .                  | 56 |
| 3.27 | Magnitud de campo $ E_{total} $ . . . . .                                | 56 |
| 3.28 | Magnitud de campo $ E_{total} $ . . . . .                                | 57 |
| 3.29 | Parte real del combinador a 20 GHz. . . . .                              | 58 |
| 3.30 | Mallado de la esfera recubierta $0.1\lambda$ . . . . .                   | 59 |
| 3.31 | RCS bistática de la esfera para un mallado de $0.1 \lambda$ . . . . .    | 60 |
| 3.32 | RCS biestática de la esfera para un mallado de $0.1 \lambda$ . . . . .   | 61 |
| 3.33 | NASA Almond . . . . .  | 62 |
| 3.34 | Descripción matemática de la NASA Almond . . . . .                       | 62 |
| 3.35 | Detalles de la excitación monoestática . . . . .                         | 63 |
| 3.36 | RCS monoestática para polarización vertical . . . . .                    | 64 |
| 3.37 | RCS monoestática para polarización horizontal . . . . .                  | 64 |
| 3.38 | RCS monoestática para polarización vertical usando IEEE . . . . .        | 65 |
| 3.39 | RCS monoestática para polarización horizontal usando IEEE . . . . .      | 65 |
| 3.40 | Bocina circular . . . . .  | 67 |
| 3.41 | Directividad de la bocina para $\Phi = 0$ usando ABC . . . . .           | 68 |
| 3.42 | Directividad de la bocina para $\Phi = 90$ usando ABC . . . . .          | 69 |
| 3.43 | Directividad de la bocina para $\Phi = 0$ usando IEEE . . . . .          | 69 |
| 3.44 | Directividad de la bocina para $\Phi = 90$ usando IEEE . . . . .         | 70 |
| A.1  | Bocina . . . . .   | 76 |
| A.2  | Boca de la Apertura . . . . .  | 76 |
| A.3  | Circunferencia . . . . .   | 77 |
| A.4  | Circunferencia con rectángulo auxiliar . . . . .                         | 77 |
| A.5  | Circunferencia con los puntos de unión . . . . .                         | 78 |
| A.6  | Boca Apertura 2 . . . . .  | 78 |
| A.7  | Apertura de la Bocina . . . . .  | 79 |

|      |  |    |
|------|--|----|
| A.8  | Guía Rectangular . . . . .                 | 80 |
| A.9  | Localización Guía Rectangular . . . . .    | 80 |
| A.10 | Parámetros Guía Rectangular . . . . .      | 81 |
| A.11 | Guía + Apertura . . . . .                  | 81 |
| A.12 | Superficies laterales . . . . .            | 81 |
| A.13 | Eliminar entidades auxiliares . . . . .    | 82 |
| A.14 | Superficies Laterales . . . . .            | 83 |
| A.15 | Geometría de la bocina circular . . . . .  | 84 |
| A.16 | Parche . . . . .                           | 85 |
| A.17 | Parche-Circunferencia . . . . .            | 86 |
| A.18 | Caja Paso 1 . . . . .                      | 86 |
| A.19 | Circunferencia + Dos Rectángulos . . . . . | 87 |
| A.20 | Circunferencia + Caja . . . . .            | 88 |
| A.21 | Eliminación Superficies . . . . .          | 89 |
| A.22 | Caja . . . . .                             | 89 |
| A.23 | Inicio del asistente. . . . .              | 90 |
| A.24 | Localización del Parche . . . . .          | 90 |
| A.25 | Parámetros del Parche . . . . .            | 91 |
| A.26 | Parche . . . . .                           | 91 |
| A.27 | Parche . . . . .                           | 91 |





---

# ÍNDICE DE TABLAS

|      |   |    |
|------|---|----|
| 1.1  | Magnitudes y parámetros de la formulación . . . . .                   | 9  |
| 3.1  | Parámetros S de la Guía Rectangular . . . . .                         | 38 |
| 3.2  | Parámetros S del Codo Plano H de $90^\circ$ . . . . .                 | 40 |
| 3.3  | Parámetros S del Codo Plano E de $90^\circ$ . . . . .                 | 42 |
| 3.4  | Parámetros S de la cascada con $d = 0$ . . . . .                      | 44 |
| 3.5  | Parámetros S de la cascada con $d = 0.5 a$ . . . . .                  | 45 |
| 3.6  | Parámetros S del cambio de ancho de la guía . . . . .                 | 47 |
| 3.7  | Parámetros S del cambio del alto de la guía . . . . .                 | 49 |
| 3.8  | Parámetros S del cambio en el alto y en el ancho de la guía . . . . . | 50 |
| 3.9  | Parámetros S de la T Mágica . . . . .                                 | 52 |
| 3.10 | Parámetros S de la unión en T con tabique centrado . . . . .          | 55 |
| 3.11 | Parámetros S del combinador . . . . .                                 | 57 |
| 3.12 | Comparativa de tiempo frente a mallado . . . . .                      | 60 |
| 3.13 | Comparativa Memoria tiempo de cómputo . . . . .                       | 61 |
| 3.14 | Comparativa Memoria tiempo de cómputo . . . . .                       | 66 |
| 3.15 | Comparativa Memoria tiempo de cómputo . . . . .                       | 70 |
| 5.1  | Fases del Proyecto . . . . .  | 74 |
| 5.2  | Coste personal . . . . .  | 74 |
| 5.3  | Costes materiales . . . . .   | 74 |
| 5.4  | Costes indirectos . . . . .   | 74 |
| 5.5  | Costes Totales . . . . .  | 74 |



---

# ÍNDICE DE PSEUDOCÓDIGOS

|      |  |    |
|------|--|----|
| 2.1  | Inicializar Ventana . . . . .                | 17 |
| 2.2  | Definir Ventana . . . . .                    | 17 |
| 2.3  | Evento cancel . . . . .                      | 18 |
| 2.4  | Evento next . . . . .                        | 18 |
| 2.5  | Finish . . . . .                             | 19 |
| 2.6  | Configurar ventanas (A) . . . . .            | 20 |
| 2.7  | Posicionamiento de las subventanas . . . . . | 20 |
| 2.8  | Configurar ventanas (B) . . . . .            | 22 |
| 2.9  | Configurar ventanas (C) . . . . .            | 22 |
| 2.10 | Definir Vectores . . . . .                   | 27 |
| 2.11 | Definir Dimensiones . . . . .                | 27 |
| 2.12 | Origen . . . . .                             | 27 |
| 2.13 | Dibujar líneas . . . . .                     | 28 |
| 2.14 | Dibujar superficies . . . . .                | 28 |
| 2.15 | Definir Puertos y Volumen . . . . .          | 28 |
| 2.16 | Dibujar tapa puerto . . . . .                | 29 |
| 2.17 | Definir Parámetros . . . . .                 | 32 |
| 2.18 | Definir Centros . . . . .                    | 32 |
| 2.19 | Origen . . . . .                             | 33 |
| 2.20 | Eliminar Contornos . . . . .                 | 33 |
| 2.21 | Dibujar superficies . . . . .                | 34 |



---

---

# CAPÍTULO 1

---

## INTRODUCCIÓN

### 1.1 Antecedentes del proyecto

A la hora de realizar el análisis de las propiedades electromagnéticas tanto de estructuras guiadas como no guiadas, o al hacer estudios de campos electromagnéticos en el interior o en el exterior de estas estructuras, surgen dos dificultades de, a priori, difícil solución.

La primera de ellas, es plantear la formulación del problema para poder resolverlo de forma teórica. Esta formulación es muy compleja y llegar a una solución correcta requiere un gran trabajo y en la mayoría de las ocasiones no es abarcable en términos de cómputo y complejidad. Por contra, si en lugar de decidirse por la solución teórica, se trata de resolver por medio de una solución experimental, nos encontramos con una dificultad añadida. La implementación física de estas estructuras para observar el comportamiento electromagnético real sería un trabajo inabarcable en términos económicos y de complejidad. Es por esto que es necesaria la utilización de herramientas software que permitan llevar a cabo la simulación de los problemas bajo análisis y de esta manera poder obtener resultados lo más precisos posible de forma mucho más sencilla y económica.

Típicamente, el análisis de problemas electromagnéticos requiere de formulaciones complejas cuya solución se quiere obtener de la forma más precisa posible. Muchas de estas ecuaciones no poseen una solución en forma de ecuaciones elementales, o son excesivamente complejas. Para poder resolverlas, las herramientas software actuales hacen uso de alguno de los métodos numéricos existentes intentando simplificar el análisis de dichas ecuaciones.

Un método numérico es un procedimiento por el cual se obtiene la solución de determinados problemas mediante la realización de cálculos lógicos y aritméticos. Están formados por una serie de instrucciones que determinan una secuencia de operaciones algebraicas y lógicas y que llevan a la obtención de una solución aproximada del problema.

Los softwares electromagnéticos presentes en el mercado se basan, sobre todo, en tres grandes métodos numéricos puros como son el Método de los Momentos (MoM) [1, 2, 3] el Método de los Elementos Finitos (FEM) [4, 5, 6] y el Método de las Diferencias Finitas en el dominio del tiempo (FDTD) [7, 8, 9]. Bajo este marco, una de las principales líneas de trabajo del grupo de investigación para el cual he colaborado en la realización de este Proyecto Fin de Carrera (PFC), ha sido el desarrollo durante años de diversos códigos electromagnéticos basados en FEM.

FEM ha demostrado ser una herramienta de cálculo numérico precisa y de gran flexibilidad. Su aplicación a problemas electromagnéticos data de la década de los sesenta, experimentando un espectacular avance durante los últimos años. Su gran ventaja estriba en su flexibilidad, derivada del propio fundamento del método. Ofrece también una gran precisión, siempre que la discretización sea suficientemente elevada e inteligente, para lo que se emplean técnicas de mallado autoadaptativo entre otros procedimientos. Como inconveniente cabe citar el tamaño de las matrices resultantes. En este sentido, es muy importante aprovechar, de forma eficiente, la estructura dispersa de las matrices, fruto del carácter local del operador diferencial. De este modo, pueden abordarse problemas complejos, manteniendo los requerimientos de almacenamiento y tiempos de cálculo dentro de unos límites razonables.

En una breve definición, el Método de los Elementos Finitos consiste en la división de un continuo, en un conjunto de pequeños elementos interconectados por una serie de puntos llamados nodos. Las ecuaciones que rigen el comportamiento del continuo, regirán también el del elemento. De esta forma, se consigue pasar de un sistema continuo (infinitos grados de libertad), que es regido por una ecuación diferencial o un sistema de ecuaciones diferenciales, a un sistema con un número de grados de libertad finito, cuyo comportamiento se modela por un sistema de ecuaciones lineales o no. De este modo, la solución de un problema complejo, se obtiene mediante el análisis de problemas "sencillos", es decir, dentro de cada uno de los elementos [10]. Como ya se ha resaltado, esto confiere una gran flexibilidad al método, lo que permite que cualquier estructura pueda ser analizada mediante FEM, sin importar la forma ni la disposición.

Como se ha comentado previamente, una de las principales líneas de trabajo del grupo de investigación para el cual he colaborado con este PFC, ha sido el desarrollo de códigos electromagnéticos basados en FEM. Durante los últimos 7 años, el trabajo se ha centrado en la implementación de un herramienta software completa capaz de realizar el análisis de radiación de antenas, el cálculo de la sección radar (RCS) de diversos objetos o el análisis de estructuras de guías de onda de manera sencilla y precisa [11]. Este herramienta software además de tener el citado módulo de FEM que realiza el cálculo del problema, posee una interfaz gráfica de usuario (GUI) que intenta ofrecer una forma amigable y sencilla de uso.

Existen dos aspectos críticos para lograr que una herramienta software de este tipo pueda ser utilizada por investigadores. El primero de ellos es realizar un exhaustivo proceso de validación/verificación de sus resultados. Además, para que la curva de aprendizaje inherente en estas herramientas sea lo mas rápida posible es necesario que tenga una interfaz gráfica de usuario sencilla y, sobre todo, muy fácil de usar.

Es, por tanto, en estos dos puntos donde se centra la realización del presente PFC. Por un lado pretende proporcionar elementos que ayuden a mejorar la usabilidad de la GUI de la herramienta, y por otra parte valide los resultados dados por la misma obteniendo conclusiones que ayuden a los investigadores a realizar simulaciones lo más precisas en el menor tiempo posible. Los detalles de estos objetivos son descritos de forma detallada en la siguiente sección.

## 1.2 Objetivos

El primer objetivo del presente PFC consiste en mejorar la interfaz gráfica de usuario que posee el código FEM desarrollado en el grupo de investigación de la universidad para que sea más manejable y permita el modelado de estructuras complejas de manera fácil y sencilla. Por este motivo, se ha creado un asistente o "*wizard*" que facilita la creación de la geometría de las estructuras que excitan, por ejemplo, las antenas o guías de ondas (en otras palabras facilita la creación de los llamados puertos de excitación).

De esta forma, durante la fase de modelización de un problema, podremos recurrir al asistente para completar la excitación de las estructuras simplemente introduciendo una serie de parámetros característicos reduciendo así los tiempos en el modelado del problema y mejorando la sencillez del proceso, tal y como se detalla en el Capítulo 2.

Por otro lado, para que una herramienta software sea confiable es necesario validar sus resultados antes de que pueda ser usada tanto en un ámbito de investigación como en uno comercial. Por tanto, el segundo objetivo de este proyecto ha sido la validación de los resultados dados por el código FEM obteniendo conclusiones que ayuden a realizar simulaciones lo más precisas en el menor tiempo posible.

Con este propósito, se ha realizado una extensa batería de pruebas dividida en dos grandes grupos de problemas como son: problemas de estructuras cerradas y problemas de estructuras abiertas o radiación. Dentro de cada uno de estos grupos se han simulado diferentes problemas con el objetivo de probar las máximas partes del código posible para minimizar el riesgo de encontrar errores en un futuro. Estas pruebas han sido perfectamente estructuradas y ejecutadas con el rigor oportuno, tal como veremos en el Capítulo 3. Además, y como objetivo secundario, se ha propuesto crear un documento que recopile numerosas pruebas y que sirva para la comprobación de códigos que se puedan desarrollar en un futuro dentro del grupo de investigación.

### 1.3 Software utilizado

En esta sección se va a dar una pequeña explicación de los componentes software utilizados en la implementación de este PFC. La GUI existente dentro del grupo de investigación está basada en el preprocesador de propósito general llamado GiD[12]. De este modo, la implementación del asistente de ayuda al usuario (primer objetivo de este PFC) ha sido realizada utilizando este software comercial de preparación de datos para simulaciones (véase la Sección 1.3.1 para más detalles sobre GiD). Con respecto al segundo objetivo de este PFC, como es la validación de los resultados del código FEM, el software utilizado, como es de esperar, es el propio código de simulación [11] (véase la Sección 1.3.2 para obtener más detalles).

#### 1.3.1 GiD

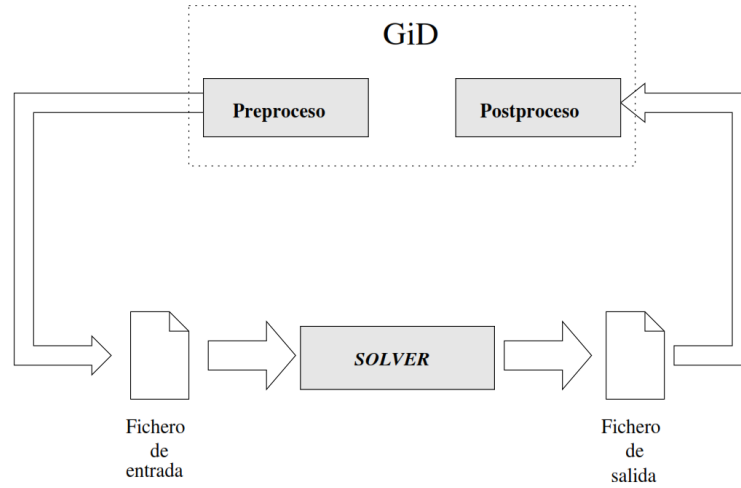
GID es un pre-post procesador de propósito general que, generalmente, se utiliza para la preparación de los datos de entrada de simulaciones, y la posterior visualización de los resultados. Ha sido concebida para cubrir las necesidades más comunes a la hora de realizar simulaciones, tales como el modelado de las geometrías, definición de los datos, mallados, transferencia de datos a un software de análisis y la posterior visualización de los datos obtenidos.

Uno de los principales argumentos para la utilización de GID, es su adaptabilidad, ya que se adapta a cualquier código de simulación numérica. GID es capaz de leer y escribir datos en multitud de formatos y permite importar y exportar el mallado o la geometría en la mayoría de los formatos usados por los principales software comerciales de CAD (Computer Aided Design), como por ejemplo DXF, VRML, IGES, ACIS, 3D Studio o Rhinoceros. Además, al ser multilenguaje, permite traducir a varios idiomas los cuadros de diálogo, los menús y las ventanas de los módulos desarrollados.

La customización de GID se realiza mediante módulos o problemas tipo (denominados en terminología anglosajona como «problem type»), que pueden ser implementados por los desarrolladores realizando una definición específica de algunas de las propiedades de las geometrías a desarrollar utilizando el lenguaje de programación TCL-TK [13, 14]. A la hora de crear un «problem type» GID permite la introducción de nuevas barras de herramientas para facilitar el acceso a los menús o la eliminación de algunas opciones que el usuario no vaya a utilizar. Estas propiedades confieren al programa una gran capacidad de adaptación a la necesidades de cada grupo de investigación.

La Fig. 1.1 muestra un diagrama de bloques en el que se puede observar como utilizar GiD para la creación de un problema tipo. El desarrollador comienza en la fase de preproceso, posteriormente se genera el fichero de entrada al código de simulación, se calcula la solución al problema generando los fichero de salida, que finalmente se visualizan en la fase de postproceso. Los detalles sobre las modificaciones realizadas por este PFC en las diferentes fases de creación de un problema tipo se muestran a continuación.





**Figura 1.1:** Diagrama de bloques

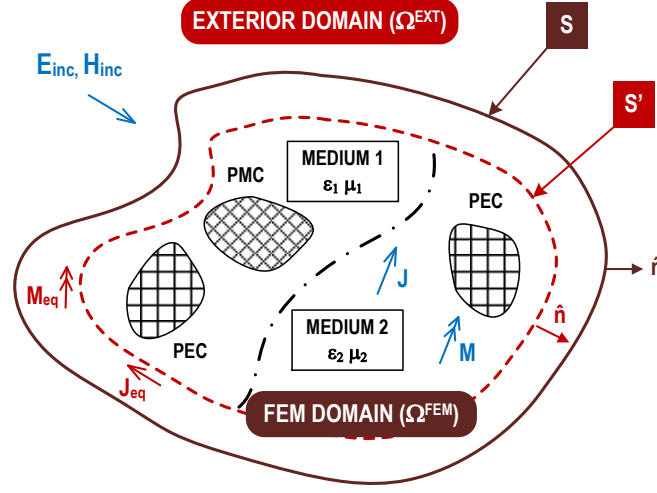
- **PREPROCESO:** Durante la fase de preprocesado, se realiza la definición de la geometría que se quiere simular, así como las condiciones de contorno, los materiales que la componen y otros parámetros relevantes. La herramienta realiza un mallado y transfiere de esta forma los datos asociados a la geometría.

En este punto se puede definir la geometría manualmente a través de GiD, o se puede utilizar el asistente desarrollado en este PFC para hacer que esta fase sea más sencilla. De este modo se han codificado varios archivos en lenguaje TCL, que permiten realizar algunas partes de este diseño de forma automática, tales como la introducción de las geometrías que excitan las estructuras, lo que permitirá reducir sensiblemente el tiempo utilizado en esta etapa. Dicho asistente viene detallado en el Capítulo 2.

Posteriormente se realiza el mallado de la estructura mediante tetraedros y se definen los puertos y excitaciones del problema que queremos simular.

- **SIMULACION:** A continuación se realiza la ejecución del análisis. Para ello se utiliza el código de simulación electromagnética desarrollado en este grupo de investigación. Este código como ya se ha indicado anteriormente está basado en el Método de los Elementos Finitos.
- **POSTPROCESO:** Finalmente en la etapa de postprocesado, se realiza la visualización de los resultados de forma fácilmente interpretable por el usuario por medio de gráficas, colores, curvas de nivel, vectores y otra serie de funciones.

En nuestro caso extraeremos los parámetros de scattering (S-Parameters en terminología anglosajona) de las estructuras simuladas en forma de gráficas y representaremos tridimensionalmente los campos electromagnéticos de alguno de los problemas, tal y como puede verse en el Capítulo 3.



**Figura 1.2:** Disposición típica de un problema FE-IIIE

Este PFC se centra fundamentalmente en la fase de preproceso, con el desarrollo del mencionado asistente que mejora la interfaz gráfica, y en la fase de postproceso realizando la verificación de los resultados. Tal y como se ha comentado previamente, el Capítulo 2 muestra como se ha implementado el asistente, mientras que en el Anexo A, se detalla la forma de usarlo en forma de tutorial.

### 1.3.2 Modulo FEM

El código de simulación usado en este PFC es un código de propósito general basado en el método conocido como Finite Element - Iterative Integral Equation Evaluation (FE-IIIE). Este código usa una formulación basada en la ecuación de onda del doble rotacional.

El dominio infinito original se divide en dos dominios que se superponen: un dominio FEM ( $\Omega^{\text{FEM}}$ ) rodeado por una superficie  $S$  y un dominio infinito exterior a una superficie auxiliar  $S'$  ( $\Omega^{\text{EXT}}$ ). La región donde se superponen ambos dominios está limitada por  $S'$  y  $S$  (ver Fig. 1.2). Por simplicidad se asume que la región exterior de  $S$  es homogénea. El límite de la superficie  $S$  puede ser de una forma arbitraria pero típicamente se selecciona que sea conforme a  $S'$ . La distancia entre  $S'$  y  $S$  generalmente es pequeña. De este modo, el dominio FEM puede ser truncado muy cerca de las fuentes para reducir el número de incógnitas del problema.

El dominio finito interior, es modelado mediante el método de los elementos finitos (FEM) usando la condición de contorno de Cauchy para el truncamiento de la malla. El sistema algebraico de ecuaciones que caracteriza el dominio interior  $\Omega^{\text{FEM}}$  puede obtenerse en términos del campos eléctrico ( $\mathbf{E}$ ) o magnético ( $\mathbf{H}$ ) a través de,

$$\nabla \times \left( \bar{\bar{f}}_r^{-1} \nabla \times \mathbf{V} \right) - k_0^2 \bar{\bar{g}}_r \mathbf{V} = \mathbf{q} \quad (1.1)$$

**Tabla 1.1:** Magnitudes y parámetros de la formulación

|          |          |                    |                    |                  |          |           |                       |                       |
|----------|----------|--------------------|--------------------|------------------|----------|-----------|-----------------------|-----------------------|
|          | <b>V</b> | $\bar{f}_r$        | $\bar{g}_r$        | $h$              | <b>O</b> | <b>L</b>  | $\Gamma_D$            | $\Gamma_N$            |
| <b>E</b> | <b>E</b> | $\bar{\mu}_r$      | $\bar{\epsilon}_r$ | $\eta$           | <b>J</b> | <b>M</b>  | $\Gamma_{\text{PEC}}$ | $\Gamma_{\text{PMC}}$ |
| <b>H</b> | <b>H</b> | $\bar{\epsilon}_r$ | $\bar{\mu}_r$      | $\frac{1}{\eta}$ | <b>M</b> | <b>-J</b> | $\Gamma_{\text{PMC}}$ | $\Gamma_{\text{PEC}}$ |

donde  $k_0$  es el número de onda en el vacío y  $\mathbf{q}$  es la fuente debida a la presencia de las corrientes eléctricas y/o magnéticas en  $\Omega^{\text{FEM}}$ . La Tabla 1 muestra las diferentes magnitudes involucradas en las formulaciones de **(E)** o **(H)**.

$$\mathbf{q} = -j k_0 \eta_0 \mathbf{O} - \nabla \times \left( \bar{f}_r^{-1} \mathbf{L} \right) \quad (1.2)$$

Además de la ecuación diferencial dada en (1.1), se necesita definir las condiciones de contorno en las superficies que delimitan el dominio interior  $\Omega^{\text{FEM}}$  para tener una única solución electromagnética. Las ecuaciones de estas condiciones de contorno son las siguientes,

$$\hat{\mathbf{n}} \times \mathbf{V} = 0 \quad \text{over } \Gamma_D \quad (1.3)$$

$$\hat{\mathbf{n}} \times \left( \bar{f}_r^{-1} \hat{\mathbf{n}} \times \mathbf{V} \right) = 0 \quad \text{over } \Gamma_N \quad (1.4)$$

$$\hat{\mathbf{n}} \times \left( \bar{f}_r^{-1} \hat{\mathbf{n}} \times \mathbf{V} \right) + \gamma \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V} = \Psi \quad \text{over } \Gamma_C \quad (1.5)$$

siendo  $\Gamma_D$ ,  $\Gamma_N$  y  $\Gamma_C$  los límites donde las condiciones de contorno de Dirichlet, Neumann y Cauchy son aplicadas respectivamente. El símbolo  $\hat{\mathbf{n}}$  es el vector unitario saliente a la superficie de contorno considerada. Las condiciones de contorno anteriores son típicamente usadas para modelar límites eléctricos y magnéticos perfectos, excitaciones mediante puertos, e incluso condiciones de absorción en problemas de radiación. El símbolo  $\gamma$  denota la constante de propagación compleja del correspondiente modo del puerto u onda exterior.

La formulación usada por el código a la hora de resolver el problema electromagnético viene dada por

$$c(\mathbf{F}, \mathbf{V}) = l(\mathbf{F}), \quad \forall \mathbf{F} \in \mathbf{W} \quad (1.6)$$

donde las fórmulas bilineal y lineal,  $c(\mathbf{F}, \mathbf{V})$  y  $l(\mathbf{F})$ , se definen como,

$$c(\mathbf{F}, \mathbf{V}) = \iiint_{\Omega} \nabla \times \mathbf{F} \cdot \left( \bar{f}_r^{-1} \nabla \times \mathbf{V} \right) dV - k_0^2 \iiint_{\Omega} \mathbf{F} \cdot \bar{g}_r \mathbf{V} dV + \gamma \iint_{\Gamma_C} (\hat{\mathbf{n}} \times \mathbf{F}) \cdot (\hat{\mathbf{n}} \times \mathbf{V}) dS \quad (1.7)$$

$$l(\mathbf{F}) = \underbrace{\iiint_{\Omega} \mathbf{F} \cdot \mathbf{q} dV}_{b_I} - \underbrace{\iint_{\Gamma_C} \mathbf{F} \cdot \Psi dS}_{b_{\Psi}} \quad (1.8)$$

La discretización de la fórmula variacional anterior se consigue usando versiones propias de funciones base isoparamétricas de alto orden que el grupo de investigación ha desarrollado durante los últimos años [5, 15, 16]. Es importante remarcar que las funciones base son obtenidas en los elementos de referencia o maestro y son transformadas al elemento real usando la inversa de la matriz Jacobiana.

Los elementos tetraédricos son muy versátiles en la discretización de dominios complejos, sin embargo, la generación de la malla usando estos elementos pueden ser muy caras hablando en términos computacionales, o simplemente no apropiadas o ineficientes para un significativo número de modelos. Este es el caso de las antenas planas en las cuales la malla volumétrica en términos de elementos prismáticos de base triangular puede ser fácilmente generada por extrusión de una malla de dos dimensiones basada en elementos triangulares. Por lo tanto, para generar simulaciones eficientes de estas antenas, el código maneja mallas volumétricas basadas en ambos tipos de elementos, los tetraédricos y los prismas de base triangular, reduciendo el tamaño del problema en comparación con mallas basadas solo en tetraedros.

Cuando se analizan problemas de radiación, el código implementa un método de truncamiento de malla llamado Iterative Integral Equation Evaluation Method (IIEE). El método proporciona una condición de contorno arbitraria y precisa (asintóticamente exacta con el número de iteraciones) mientras la estructura dispersa en la matriz se mantiene. Esta técnica reduce considerablemente el tamaño del problema truncando el dominio FEM muy cerca de las fuentes (típicamente en el rango de  $0.05\lambda$  a  $0.2\lambda$ ). Por lo tanto, el código puede reducir los requerimientos computacionales para el análisis de estructuras grandes (con la correspondiente mejora en el tiempo de simulación) usando diferentes tipos de elementos finitos y el método IIEE para el truncamiento de mallas.

## 1.4 Contenido del Proyecto

Este Proyecto Fin de Carrera está estructurado de la siguiente manera:

- Capítulo 1: Introducción

En este capítulo se realiza, por un lado la explicación de los motivos por los que se ha llevado a cabo este PFC (Sección 1.1), y por otro una relación de los objetivos que se pretenden conseguir en el mismo (Sección 1.2). Además, se presentan las herramientas utilizadas a la hora de su implementación (Sección 1.3)

- Capítulo 2: Asistente para la creación de modelos

En este capítulo se realiza una descripción detallada del asistente o "wizard". La descripción se centra en los motivos que han llevado a su desarrollo, las funcionalidades de las que dispone y como han sido implementadas dentro del asistente.

- Capítulo 3: Verificación de Resultados

En este capítulo se detallan todas las pruebas de verificación ejecutadas a la hora de comprobar el correcto funcionamiento del código de elementos finitos. Estas pruebas se dividen en problemas de estructuras cerradas en tecnología de guías de onda y problemas de radiación.

- Capítulo 4: Conclusiones

En este capítulo se exponen las conclusiones obtenidas durante la realización del presente PFC, así como, se mencionan algunas posibles futuras mejoras y líneas de investigación.

- Capítulo 5: Presupuesto

En este capítulo se realiza un cálculo lo más aproximado posible de los costes económicos de la realización de este PFC. Para este cálculo se tienen en cuenta tanto los costes materiales como los costes de personal.

- Anexo A: Manual

En este anexo se adjunta un manual detallado que muestra al usuario como utilizar de forma sencilla, el asistente desarrollado en este PFC. El manual describe como crear dos estructuras básicas de microondas utilizando dicho asistente.



---

---

# CAPÍTULO 2

---

## ASISTENTE PARA LA CREACIÓN DE MODELOS

### 2.1 Introducción

Como se ha comentado en el Capítulo 1, a la hora de desarrollar un código de computación, no basta con que dicho código sea eficiente y preciso, sino que además debe ser sencillo de manejar. Por ello, uno de los principales objetivos en la creación de nuevos códigos de simulación es que posean una interfaz gráfica de usuario amigable, sencilla y muy fácil de usar.

El código de elementos finitos presentado anteriormente, posee una interfaz de usuario que permite la creación de estructuras electromagnéticas complejas mediante la utilización de las llamadas superficies NURBS [17]. Gracias a estas superficies es posible crear estructuras tan complejas como vehículos, antenas o circuitos de microondas con materiales inhomogéneos desde cero. El principal problema que presenta esta forma de crear los modelos es que la curva de aprendizaje necesaria para el modelado de dichas estructuras es alta.

Como se ha comentado en la introducción, GID puede ser customizado y de este modo tener una apariencia y operatividad diferente. Dentro de estas posibilidades que ofrece GID, se ha decidido desarrollar un asistente con el objetivo de reducir el tiempo a la hora de crear los diferentes modelos de estructuras. Este asistente facilitará la definición de la geometría de los puertos que excitan dichas estructuras. Esta tarea es compleja y tediosa, por lo que con el uso del asistente reduciremos sensiblemente el tiempo en

el modelado. En particular, el asistente ofrece la posibilidad de generar los puertos de estructuras excitadas mediante guíasondas (filtros, estructuras de transmisión de alta potencia, etc...) o mediante un cable tipo coaxial (como por ejemplo, circuitos o antenas fabricados en tecnología planar).

Un asistente se define como una interfaz gráfica de usuario, que presenta información y posibilita la realización de tareas correlativamente a través de diferentes pasos. Estos pasos pueden variar en función de las elecciones que realiza el usuario. Para poder crear este asistente, se han codificado diversos archivos en el lenguaje de script **TCL-TK** (lenguaje en el que está escrito **GID**). De esta manera al invocar el asistente, se genera una ventana que permite crear la geometría de los puertos que excitan los dispositivos deseados. Esta ventana va solicitando algunos parámetros que se introducen por medio del teclado o seleccionando con el ratón las opciones correspondientes.

El asistente utiliza los paquetes «**Tkwizard**» y «**BWidget**» que configuran su base y aportan ciertas utilidades para su programación como se verá en la sección 2.2. El asistente está definido en tres pasos o «**Steps**», en el primero de los cuales se elegirá la estructura que se va a generar, en el segundo se introduce todo lo referente a la localización y posicionamiento de dicha estructura, y en el tercero y último, se introducen sus parámetros característicos y dimensiones. En cada una de estas ventanas se dispone de unos botones que permiten el poder avanzar o retroceder en los distintos pasos, y dos botones adicionales para finalizar la creación de la estructura o cancelarla por completo. El funcionamiento completo de este asistente puede verse en el Manual de Usuario adjunto en el Anexo A.

## 2.2 Implementación del asistente

En la presente sección se van a describir los diversos detalles técnicos sobre la implementación del asistente de ayuda al modelado desarrollado en este PFC. El asistente está basado en un paquete de **TCL-TK** llamado «**TKWizard**», el cual proporciona los comandos necesarios para su implementación de manera rápida y sin demasiadas complicaciones. A continuación se va a detallar un breve resumen de las funcionalidades y los principales comandos que presenta dicho paquete.

### 2.2.1 Paquete tkwizart

**TKWizard** es un conjunto de ficheros escritos en **TCL-TK** utilizado para la creación de asistentes de ayuda al usuario. Este paquete funciona en cualquier versión posterior a la 8.0 del propio lenguaje **TCL-TK** y proporciona una base común para poder crear los mencionados asistentes de forma sencilla. **TKWizard** incluye el manejo de los componentes de las ventanas y de los pasos del asistente. También incluye los diversos eventos que se producen, automatizando su creación. Por tanto, sirve como marco inicial para su implementación haciendo que no sea necesario desarrollar el código completo desde cero.



A continuación se van a describir algunos de los comandos y eventos más comunes del paquete que se utilizan en la implementación de los asistentes de ayuda al usuario:

### CONSTRUCTOR DEL ASISTENTE

Invocando al método `«tkwizard::tkwizard»`, se crea un nuevo objeto de tipo `TkWidget`. Presenta numerosas opciones en su creación, algunas de las que se han utilizado son:

- **Title:** Será el título que aparezca en la barra superior de la ventana.
- **Canceltext:** define el texto que aparecerá en el botón `«Cancel»`
- **Finishtext:** define el texto que aparecerá en el botón `«Finish»`
- **Showhelp:** muestra el botón `«Help»`

En nuestro caso, la sentencia utilizada ha sido la siguiente: `"tkwizard::tkwizard $W2 -title [= "Predefined structures"] -canceltext [= Cancel] -finishtext [= "Finish"]`. Según la cual se genera un nuevo Widget tipo wizard `$W2`, en cuya cabecera mostrará el título `"predefined structures"`. También define las etiquetas para los botones de cancelar y finalizar, en este caso serán `Cancel` y `Finish`.

### CONSTRUCTOR DE UN PASO DEL ASISTENTE

Una vez construido un objeto de tipo `TkWidget`, hay que crear cada uno de los pasos del mismo. Para ello se invoca al método `«Step»`, tantas veces como pasos vaya a tener el asistente. Entre las distintas opciones que se pueden utilizar, usaremos `«Layout»` para definir qué estilo de paso se desea. Típicamente es `«basic»` pero hay otros tipos. Para más información se puede consultar la documentación. Una sentencia típica de creación de un paso, sería: `«$W2 step step1 -layout basic»`, donde `$W2` es el objeto tipo `TkWidget` para el que se está definiendo el paso, `Step1` es el nombre asignado a este paso y `basic` la opción elegida para el estilo.

### COMANDO QUE DEVUELVE EL FRAME DONDE PINTAR EL PASO

Para poder definir la estructura de cada uno de los pasos, es necesario obtener el área donde poder pintar. Para ello se realiza la siguiente sentencia: `«set Area [$this widget clientArea]»`, donde la variable `this` está definida y tiene almacenado el nombre de la página del asistente que se está creando. El proceso `clientArea` devolverá un objeto tipo widget que contendrá el Frame donde pintar el paso. Este objeto se almacena en la variable `Area`

### DEFINIR EL ORDEN DE LOS PASOS DEL ASISTENTE

Una vez definidos todos los pasos que conformarán el asistente, es necesaria indicar la secuencia del orden de los mismos. La sentencia sería la siguiente: `«$W2 order step1 step2 step3»`, donde `$W2` es el objeto tipo `TkWidget` para el que se está definiendo el orden de los pasos, `order`, es el propio método que realiza la ordenación, y la secuencia de los pasos sería `step1 step2 step3`.

### MOSTRAR EL ASISTENTE

Una vez que se tiene creado el asistente completo, tanto en apariencia como en funcionalidad, se procede a hacerlo visible mediante el comando «**Show**», el cual dibuja y coloca el asistente en la pantalla. Para ello se obtiene la referencia del asistente que se ha creado y se invoca al método. La sentencia sería la siguiente: «**\$W2 show**»

A continuación se indican algunas de las opciones más usuales que incluye el paquete **TkWizard**, que permiten cambiar la apariencia del asistente.

- **Background:** Color principal del fondo de la ventana del asistente
- **Style:** Define el estilo del Wizard, el valor por defecto presenta una apariencia muy similar a las de Windows 2000.
- **Resizable:** permite modificar las dimensiones de las ventanas.

Como ya se ha comentado, un asistente está formado por una serie de pasos que van recopilando información y realizan tareas en base a esa información. Para poder moverse por estos pasos es necesario el uso de ciertos botones, estos son «**BackButton**», «**NextButton**», «**HelpButton**» y «**CancelButton**». Estos botones pueden deshabilitarse mediante el comando '**Disabled**'.

En la implementación del paquete **TkWizard**, cada uno de estos botones tiene asociado controladores de eventos para una o más acciones, que tienen lugar al pulsar o al soltar los botones. Estos eventos son:

- «**WizNextStep**»: Evento que tiene lugar al pulsar el botón «**Next**», invoca al siguiente paso del asistente.
- «**WizPrevStep**»: Evento que tiene lugar al pulsar el botón «**Back**», invoca al paso anterior del asistente.
- «**WizFinish**»: Evento que tiene lugar al pulsar el botón «**Finish**», finaliza el asistente invocando al método que crea la estructura seleccionada .
- «**WizCancel**»: Evento que tiene lugar al pulsar el botón «**Cancel**», finaliza el asistente destruyéndolo.
- «**WizHelp**»: Evento que tendría lugar al pulsar el botón «**Help**», en este caso este botón está deshabilitado, por lo que no se le ha dotado de funcionalidad.

Hasta ahora se han mostrado los métodos y eventos utilizados para desarrollar el asistente, así como las funcionalidades que aporta el paquete utilizado. En la sección siguiente se va a explicar en detalle como se ha llevado la implementación del mismo.

### 2.2.2 Definición del asistente

Al iniciar el asistente se invoca al método «CreateMyWizard», este método define una serie de variables para su uso posterior. El primer paso que realiza es la comprobación de si ya se ha iniciado el wizard con anterioridad. Esta comprobación se lleva a cabo en las líneas 4-6 del Pseudocódigo 2.1. En el caso de iniciarse el wizard por primera vez, se procede a crear una nueva ventana(véanse líneas 8-11 del Pseudocódigo 2.1).

```
1 proc CreateMyWizard { } {
2   ...
3   #Definición de la ventana
4   set w2 .gid.myWizard
5   #Eliminar la ventana si existe
6   if { [wininfo exists $w2] } { destroy $w2 }
7
8   #Crear el wizard
9   tkwizard::tkwizard
10  \ $w2 -title ["HOFEM predefined structures"]
11  -canceltext ["Cancel"] -finishtext ["Finish"]
12    ...
13 }
```

**Pseudocódigo 2.1:** Inicializar Ventana

```
1 switch $::tcl_platform(platform) {
2     "windows" { wm geometry $w2 530x610 }
3     "unix"    { wm geometry $w2 560x760 }
4 }
5 #Iniciar Barra de Progreso
6 set progressbar 0
7
8 #Eventos de finalizar, cancelar y próximo paso
9 bind $w2 <<WizFinish>> { WIZARD::MyFinish %W}
10
11 bind $w2 <<WizCancel>> { WIZARD::MyCancel %W}
12
13 bind $w2 <<WizNextStep>> { WIZARD::NextMyStep %W}
14
15 #Definir los pasos del wizard
16   ...
17 }
```

**Pseudocódigo 2.2:** Definir Ventana

## 2.2. IMPLEMENTACIÓN DEL ASISTENTE

---

A continuación se define el tamaño que tendrá la ventana, este tamaño variará si el programa se ejecuta en un entorno Windows o Unix (ver líneas 1-4 del Pseudocódigo 2.2). Una vez definido el tamaño del asistente, se pasa a controlar los eventos de entrada de ratón propios de su ventana principal. Estos eventos son: finalizar el asistente (Botón «Finish»), pasar al siguiente paso (Botón «Next») y cancelar la ejecución del mismo (Botón «Cancel»). Tal y como se comentó en la sección previa, una vez que tienen lugar dichos eventos, se enlazan con los comandos que se desean ejecutar para así realizar las acciones oportunas. Las líneas 8-13 del Pseudocódigo 2.2 muestra esta asignación de los comandos para el asistente del presente PFC.

El siguiente paso es gestionar los eventos. El más sencillo de todos, el botón de «Cancel», invoca al comando «MyCancel» que se encarga de liberar las variables que se hayan almacenado al ejecutar el asistente y da por finalizada la ejecución, tal y como se muestra en el Pseudocódigo 2.3. En el caso del botón de «Next», véase el Pseudocódigo 2.4.

```
1 #Invoca al método Free, encargado de liberar variables
2 proc MyCancel { win } { Codigo::Free
3 ...
4 #Si hay un centro definido, se procede a desasignarlo
5 if [info exists cent] { unset cent }
6 #Liberación de los parámetros almacenados anteriormente
7 if [info exists param] { unset param }
8 }
```

Pseudocódigo 2.3: Evento cancel

```
1 proc NextMyStep { this } {
2 #Variable que almacena la estructura seleccionada
3 global em_struture
4
5 switch $em_struture {
6 #La estructura elegida ha sido la Guíaonda
7 " Waveguide"      { WIZARD::GuiaR $c }      }
8 #La estructura elegida es el Cable Coaxial
9 " Coaxial Cable"  { WIZARD::Coaxial $c } }
10 }
11 #Si no se ha elegido ninguna estructura
12 if {$em_struture == "0"} {
13 #Mensaje de Error
14     return -code break
15 }
16 }
```

Pseudocódigo 2.4: Evento next

```
1 proc MyFinish { win } {  
2  
3 switch $em_structure$ {  
4   " Waveguide" {  
5     #Chequeo de parámetros  
6     trace remove variable param(aa) write "calcparam;#"  
7     trace remove variable param(bb) write "calcparam;#"  
8  
9     Codigo::CrearAntena $win  
10  }  
11  " Coaxial Cable" {  
12    #Chequeo de parámetros  
13    trace remove variable param(aa) write "calcparam;#"  
14    trace remove variable param(bb) write "calcparam;#"  
15  
16    Codigo::CrearAntena $win  
17  }
```

**Pseudocódigo 2.5:** Finish

Al pulsar el botón «Finish» se invoca el método «MyFinish». Este método analiza que tipo de estructura es la que se va a generar, y realiza unas comprobaciones sobre los valores de los parámetros introducidos por el usuario (véanse líneas 5-7 y 12-15 del Pseudocódigo 2.5). Si todos los parámetros son correctos, en las líneas 9 y 17 del Pseudocódigo 2.5 se invoca al método que genera la estructura correspondiente.

Una vez que se han tratado los distintos eventos, hay que definir como será cada una de las ventanas, para que tengan la apariencia deseada, si bien tendrán algunas particularidades en función del paso en el que se encuentre el usuario.

En primer lugar se configura la ventana para que se pueda expandir al arrastrar el borde con el ratón, esta configuración se lleva a cabo en las líneas 5-7 del Pseudocódigo 2.6. Adicionalmente se pinta la ventana de color verde. Anteriormente el espacio sobre el que se van insertando las distintas imágenes o botones, aparecía en blanco (ver Fig. 2.1), tras definir un color mediante las líneas 9-11 del Pseudocódigo 2.6, la apariencia es la que se muestra en la Fig. 2.2. La ventana se subdivide en otras subventanas que hay que posicionar. En este caso, definiremos tres subventanas, dos en la parte superior denominadas f1 y f2, y una tercera en la parte inferior. En la primera a su vez se definen dos filas y dos columnas, para albergar elementos como barras de scroll, etiquetas e imágenes. Las otras dos por su parte sólo albergarán una imagen y una etiqueta, por lo que sólo habría que posicionarlas. El posicionamiento de estas subventanas y su distribución interior viene definido por las líneas 1-13 del Pseudocódigo 2.7. El color del fondo de las subventanas se define en las líneas 14-17 del Pseudocódigo 2.7, resultando la apariencia final que se muestra en la Fig. 2.3.

```
1 #Definir los pasos del Wizard
2 $w2 step step1 -layout basic {
3 ...
4 #Habilitar expansión al hacer más grande la ventana
5 grid columnconfigure $c 0 -weight 1
6 grid rowconfigure $c 1 -weight 0
7
8 #Definir el color de este paso
9 $this stepconfigure -step 1
10 $c configure -bg #96C800
```

**Pseudocódigo 2.6:** Configurar ventanas (A)

```
1 #Posicionar Subventanas
2 #Dos en la parte superior
3 grid $f1 $f2 -sticky nsew -padx 10 -pady 5
4 #La primera dividida en dos filas y columnas
5 grid columnconfigure $f1 0 -weight 1
6 grid columnconfigure $f1 1 -weight 0
7 grid rowconfigure $f1 0 -weight 0
8 grid rowconfigure $f1 1 -weight 0
9 #Segunda y tercera con espacio único
10 grid columnconfigure $f2 0 -weight 0$
11 #La tercera se posiciona en la parte inferior
12 grid $c.images -sticky sew -padx 5 -pady 5 -columnspan 2
13 grid columnconfigure $c.images 0 -weight 1
14 #Definimos el color para el fondo de las subventanas
15 frame $c.frame1 -background #96C800;
16 frame $c.frame2 -background #96C800
17 frame $c.images -background #96C800
```

**Pseudocódigo 2.7:** Posicionamiento de las subventanas

En cada una de las subventanas definidas se van insertando distintos elementos que conforman la apariencia final de la ventana, tales como barras de scroll (mediante la línea 4 del Pseudocódigo 2.8) o listas de elementos (líneas 5-6 del Pseudocódigo 2.8). Además para darle operatividad a estos elementos es necesario tratar los eventos que pueden producir. Así al pulsar doble click con el ratón uno de los elementos listados, se invoca al método que hace saltar al siguiente paso del asistente (véanse líneas 7-8 del Pseudocódigo 2.8). Si por el contrario únicamente se hace click una vez sobre un elemento de la lista, se invoca el método para almacenar su valor (ver línea 9 del Pseudocódigo 2.8) y se marca dicho valor en un color diferente para que el usuario sepa en todo momento qué elemento es el que ha seleccionado (líneas 12-16 del Pseudocódigo 2.8).

## 2.2. IMPLEMENTACIÓN DEL ASISTENTE



**Figura 2.1:** Ventana del asistente básica.



**Figura 2.2:** Ventana del asistente coloreada.



**Figura 2.3:** Ventana del asistente final.

Para terminar de darle el aspecto definitivo a la ventana, se procede a poner etiquetas a cada una de las subventanas creadas, para ello en primer lugar se elige el tipo de letra, el color de la misma y el fondo de la etiqueta, se indica su posicionamiento y por último se inserta la imagen que se quiere mostrar (véanse líneas 1-12 del Pseudocódigo 2.9). Este proceso se repite para cada una de las subventanas con variaciones mínimas, sólo falta indicar en qué orden han de ejecutarse los pasos del asistente (ver línea 14 del Pseudocódigo 2.9).

## 2.2. IMPLEMENTACIÓN DEL ASISTENTE

```
1 #Definir los pasos del Wizard
2 $w2 step step1 -layout basic {
3
4 #Declaración de elementos
5 scrollbar $f1.s1 -command "$f1.l1 yview"
6 listbox $f1.l1 -yscroll "$f1.s1 set" -bd 2 -relief ridge
7     -state normal -width 25
8 bind $f1.l1 <Double-ButtonPress-1> { event generate
9     .gid.turuWizard <<WizNextStep>> }
10 bind $f1.l1 <ButtonRelease-1> { WIZARD::setLabel }
11 $f1.l1 insert 0 " Waveguide" " Coaxial Cable"
12
13 switch $em_struture {
14     0 { $f1.l1 selection set 0 }
15         " Waveguide" { $f1.l1 selection set 0 }
16         " Coaxial Cable" { $f1.l1 selection set 1 }
17     }
18 }
```

Pseudocódigo 2.8: Configurar ventanas (B)

```
1 #Definir tipo de letra
2 set bfont [concat [font actual [.gid.panel.tree cget -font]]
3     [list -size 9] [list -weight bold]]
4
5 #Definir formato de la etiqueta
6 label $f2.label -text "Structure preview"-bd 2-relief groove
7 -anchor c -background #7F3400 -foreground white -font $bfont
8
9 #Colocación
10 grid $f2.label -sticky nsew -ipady 3 -pady 4
11
12 #Insertar imagen
13 label $f2.image -image [GetImage [file join $PrivDavid(img)
14     step1_all.png]] -relief groove -bd 2 -width 280 -height 160
15 grid $f2.image -sticky nsew
16
17 #Definir el orden de los pasos del wizard
18 $w2 order step1 step2 step3$
19 }
```

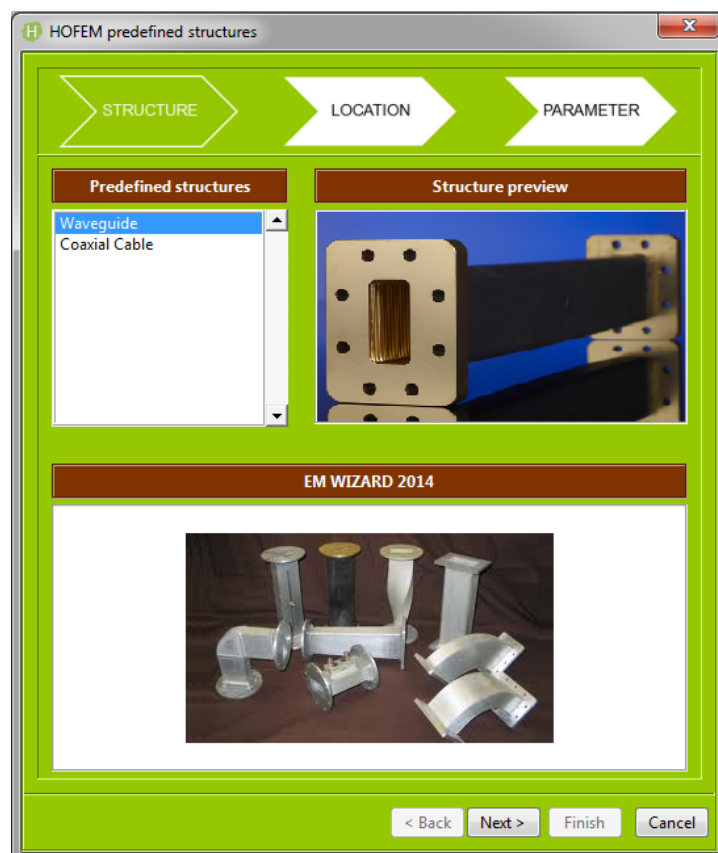
Pseudocódigo 2.9: Configurar ventanas (C)



## 2.3 Definición de los Pasos del Asistente

En la sección 2.2.2, se ha detallado como se ha llevado a cabo el desarrollo o programación de las características generales del asistente utilizando como base el paquete TkWizard 2.2.1. Este desarrollo ha incluido, por un lado la funcionalidad del asistente y de sus componentes y por otro la apariencia de cada uno de los pasos. En las dos secciones siguientes, se muestra en detalle como se han implementado cada una de las ventanas para la creación de dos formas de excitación básicas, como son la Guía Rectangular (ver 2.3.1) y el Cable Coaxial (ver 2.3.4).

Para poder iniciar el asistente es necesario pulsar el botón con forma de varita mágica que se encuentra en la parte superior derecha de la pantalla, tal como veremos en el manual de usuario del Anexo A. En este momento dará comienzo el asistente y mostrará una pantalla donde se seleccionará con el ratón la opción que se desea generar de entre las mostradas en la parte izquierda (véase la Fig. 2.4). La imagen de la derecha irá cambiando en función del tipo de excitación. Una vez seleccionada la estructura, es necesario pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso. Si por el contrario desea abandonar el asistente, es necesario pulsar el botón «Cancel» situado en la parte inferior derecha.



**Figura 2.4:** Selección de estructuras.

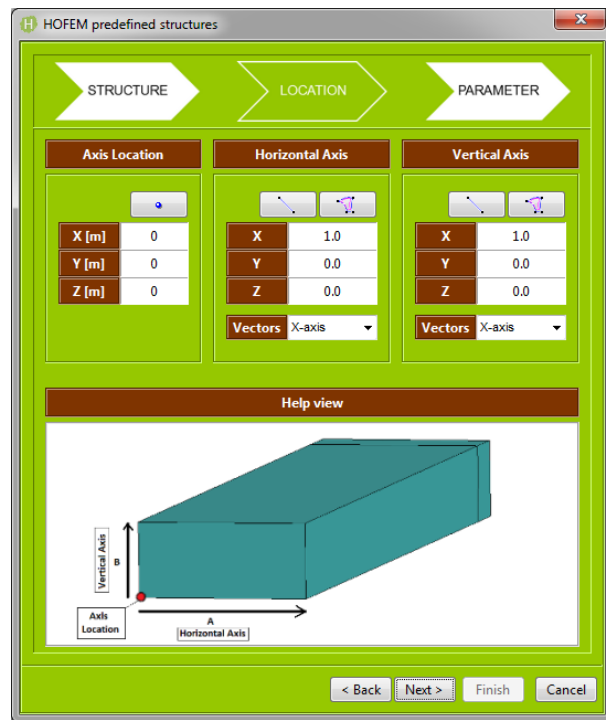



Figura 2.5: Localización de la estructura.

### 2.3.1 Guía Rectangular: Localización

La primera forma de proporcionar la excitación que se puede generar con este asistente es una guía rectangular. Es probablemente la estructura más simple pero a la vez una de las más utilizadas. La Fig. 2.5 muestra el siguiente paso del asistente para las guías rectangulares. Está dividido en dos secciones, una en la parte superior, denominado 'waveguide location' donde solicita los primeros parámetros necesarios y otra en la parte inferior, denominada 'Help view', donde se representa a modo de ayuda una imagen detallada de la guía rectangular.

Para poder generarla correctamente, el asistente solicita en primer lugar el punto de origen desde el que se empezará a dibujar la guía, denominado 'Axis Location', se introducirá de este modo las coordenadas  $X, Y, Z$  del punto de origen (véase la Fig. 2.6) También existe la posibilidad de seleccionar un punto ya existente de una estructura anterior, pulsando el botón 'Pick' .


A continuación, hay que introducir los vectores que definirán la situación de la puerta de la guía. Para ello se solicitan las coordenadas  $\langle X, Y, Z \rangle$  de dichos vectores. El asistente cuenta con un desplegable en el que poder seleccionar uno de los vectores predefinidos, que serán  $\langle X, Y, Z, -X, -Y, -Z \rangle$  (véase la Fig 2.7). El eje longitudinal de la guía vendrá definido por el producto vectorial de los dos vectores que conforman la puerta. Adicionalmente, pinchando en el botón 'Pick'  se puede seleccionar un vector ya existente de una estructura anterior.

Figura 2.6: Localización del Eje.

Figura 2.7: Situación de la puerta.

Una vez introducidas las coordenadas del punto origen y los vectores que conforman la puerta de la guía, es necesario pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso. A la izquierda de este botón, se encuentra otro denominado «Back» con el que regresar al paso anterior para seleccionar una estructura diferente. Si por el contrario se desea abandonar el asistente, es necesario pulsar el botón «Cancel» situado en la parte inferior derecha.

### 2.3.2 Guía Rectangular: Parámetros

La Fig. 2.8 muestra el último paso del asistente para las guías rectangulares. En este punto se solicitan los parámetros básicos de una guía de onda, como son la anchura, la altura y la longitud, representadas por 'Parameter(A)', 'Parameter(B)' y 'Parameter(L)'. Se dispone de dos ventanas de ayuda, representadas como «Help View» que muestran a qué corresponden cada uno de los parámetros solicitados. Una vez introducidos todos los parámetros, pulsando en el botón «Finish», se procede a la creación de la Guía Rectangular conforme a los datos que se han ido introduciendo.

### 2.3.3 Guía Rectangular: Creación

Hasta este punto es donde el usuario puede ir interactuando con el asistente, eligiendo como quiere construir la guía rectangular conforme a los parámetros que se han ido introduciendo. A partir de aquí se va a exponer como se lleva a cabo la creación de la estructura.

### 2.3. DEFINICIÓN DE LOS PASOS DEL ASISTENTE

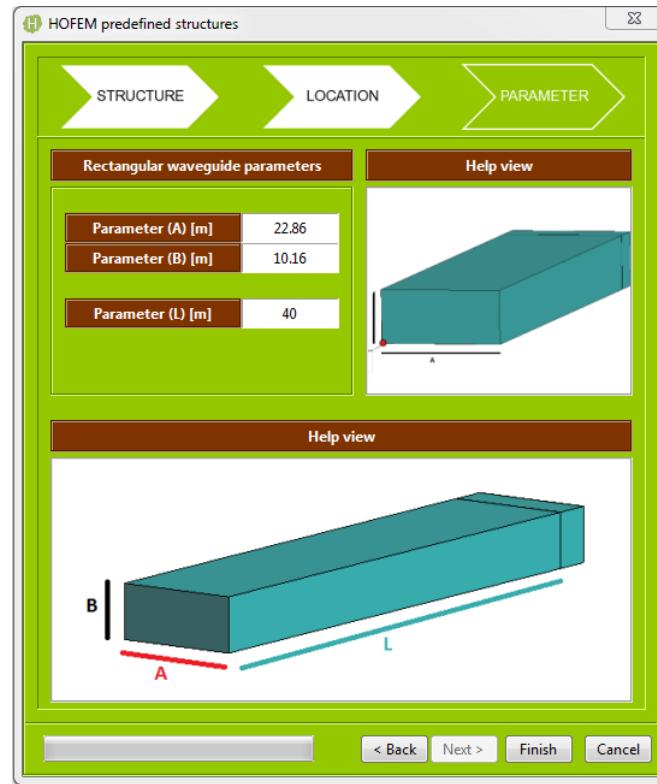


Figura 2.8: Parámetros de la estructura.

Tras pulsar el botón «Finish» se invoca el proceso 'GuiaRectangular'. En este proceso, en primer lugar, se realiza una normalización de los vectores que conforman la puerta de la guía (véanse líneas 4-7 del Pseudocódigo 2.10). A continuación, se realiza el producto vectorial de estos vectores, para de esta forma obtener el vector que definirá el eje longitudinal de la misma (líneas 10-15 del Pseudocódigo 2.10). Después se definen las medidas para el alto, ancho y longitud de la guía rectangular, tal como puede verse en las líneas 2-4 del Pseudocódigo 2.11. De forma análoga se establece el centro desde el cual se comienza a dibujar la guía mediante las líneas 6-8 del Pseudocódigo 2.11. Finalmente se procede a dibujar el primer punto, que coincide con el centro escogido (ver línea 2 del Pseudocódigo 2.12). A partir de este primer punto se van dibujando los ocho puntos totales, para ello se suman los valores de los lados y la longitud según el eje que corresponda (véanse líneas 4-7 del Pseudocódigo 2.12).

Una vez creados todos los puntos, se procede a dibujar las líneas que conforman las aristas de la guía rectangular. Para ello se van uniendo los puntos dibujados anteriormente según corresponda, como muestran las líneas 2-5 del Pseudocódigo 2.13. En este momento ya se tiene el contorno de la guía rectangular perfectamente dibujado. A continuación, se definen las superficies que vienen delimitadas por las líneas creadas anteriormente (véase líneas 2-6 del Pseudocódigo 2.14). Una vez que se han dibujado las superficies laterales de la guía, hay que generar las superficies que constituyen los puertos de la guía rectangular (véanse líneas 2-7 del Pseudocódigo 2.15).

### 2.3. DEFINICIÓN DE LOS PASOS DEL ASISTENTE

```
1 proc GuiaRectangular { } {
2
3 #Normalización de los vectores que conforman la puerta
4 set nn [MathUtils::VectorNormalized \
5         [list $vect(x) $vect(y) $vect(z)]]
6 set nn2 [MathUtils::VectorNormalized \
7          [list $vect2(x) $vect2(y) $vect2(z)]]
8 #Cálculo del vector correspondiente al eje longitudinal
9 set vect1(x) [expr [lindex $nn 1]*[lindex $nn2 2]
10              -[lindex $nn 2]*[lindex $nn2 1]]
11 set vect1(y) [expr [lindex $nn 2]*[lindex $nn2 0]
12              -[lindex $nn 0]*[lindex $nn2 2]]
13 set vect1(z) [expr [lindex $nn 0]*[lindex $nn2 1]
14              -[lindex $nn 1]*[lindex $nn2 0]]
15 ...
16 }
```

Pseudocódigo 2.10: Definir Vectores

```
1 #Definición de las dimensiones de la guía
2 set lado_a [expr $param(aa)]
3 set lado_b [expr $param(bb)]
4 set lado_l [expr $param(ll)]
5
6 #Definición del punto origen sobre el que se dibuja la guía
7 set cent_x [expr $cent(x)]
8 set cent_y [expr $cent(y)]
9 set cent_z [expr $cent(z)]
```

Pseudocódigo 2.11: Definir Dimensiones

```
1 #Dibujar el punto origen de la guía
2
3 set punto1 [GiD_Geometry create point append Layer0
4             $cent_x $cent_y $cent_z]
5 set punto2 [GiD_Geometry create point append Layer0
6             [expr $cent_x+$lado_a*$vect(x)]
7             [expr $cent_y+$lado_a*$vect(y)]
8             [expr $cent_z+$lado_a*$vect(z)]]
```

Pseudocódigo 2.12: Origen

### 2.3. DEFINICIÓN DE LOS PASOS DEL ASISTENTE

---

```
1 #Dibujar las líneas que conforman la guía
2 set linea1 [GiD_Geometry create line \
3     append stline Layer0 $punto1 $punto2]
4
5 set linea2 [GiD_Geometry create line \
6     append stline Layer0 $punto1 $punto3]
```

**Pseudocódigo 2.13:** Dibujar líneas

```
1 #Definir las superficies laterales
2 GiD_Process Geometry create NurbsSurface \
3     $linea1 $linea5 $linea6 $linea7
4 GiD_Process Mescape
5
6 GiD_Process Geometry create NurbsSurface \
7     $linea3 $linea10 $linea11 $linea12
8 GiD_Process Mescape
```

**Pseudocódigo 2.14:** Dibujar superficies

```
1 #Definición de las superficies de los puertos.
2 GiD_Process Geometry create NurbsSurface \
3     $linea1 $linea2 $linea3 $linea4
4 GiD_Process Mescape
5 GiD_Process Geometry create NurbsSurface \
6     $linea7 $linea8 $linea9 $linea10
7 GiD_Process Mescape
8
9 #Creación del volumen
10 GiD_Process Geometry create Volume \
11     [expr $superficies+1] [expr $superficies+2] \
12     [expr $superficies+3] [expr $superficies+4] \
13     [expr $superficies+5] [expr $superficies+6]
14 GiD_Process Mescape
```

**Pseudocódigo 2.15:** Definir Puertos y Volumen

Posteriormente se define un volumen englobado por todas las superficies generadas (Véanse líneas 11-14 del Pseudocódigo 2.15). Una vez que se tiene definida la guía completa, para los problemas de radiación, se cubre el extremo del puerto, de esta forma queda aislado del exterior y evita interferencias. El Pseudocódigo 2.16 muestra la definición del aislamiento del puerto.

```

1 #Definición del punto donde irá la tapa.
2 set cent3_x [expr $cent2_x + 0.5*$lado_a*$vect1(x)]
3 set cent3_y [expr $cent2_y + 0.5*$lado_a*$vect1(y)]
4 set cent3_z [expr $cent2_z + 0.5*$lado_a*$vect1(z)]
5 #Trasladar el puerto a la distancia fijada.
6 GiD_Process Mescape
7 GiD_Process Utilities Copy Lines DoExtrude Surfaces
8     Translation $cent2_x $cent2_y $cent2_z $cent3_x
9     $cent3_y $cent3_z [expr $lineas+7] [expr $lineas+8]
10    [expr $lineas+9] [expr $lineas+10]
11 #Definición de la superficie.
12 GiD_Process Mescape
13 GiD_Process Geometry create NurbsSurface [expr $lineas+13]
14    [expr $lineas+14] [expr $lineas+15] [expr $lineas+16]
15 GiD_Process Mescape$



```

Pseudocódigo 2.16: Dibujar tapa puerto

### 2.3.4 Cable Coaxial: Localización

Tras dar comienzo al asistente de nuevo, seleccionaremos con el ratón la segunda de las estructuras que se pueden seleccionar en el cuadro de la izquierda (véase la Fig. 2.9). La imagen de la derecha mostrará ahora un cable coaxial. Una vez seleccionada la estructura, es necesario pulsar el botón «Next» para avanzar al siguiente paso. Si por el contrario desea abandonar el asistente, es necesario pulsar el botón «Cancel» situado en la parte inferior derecha.

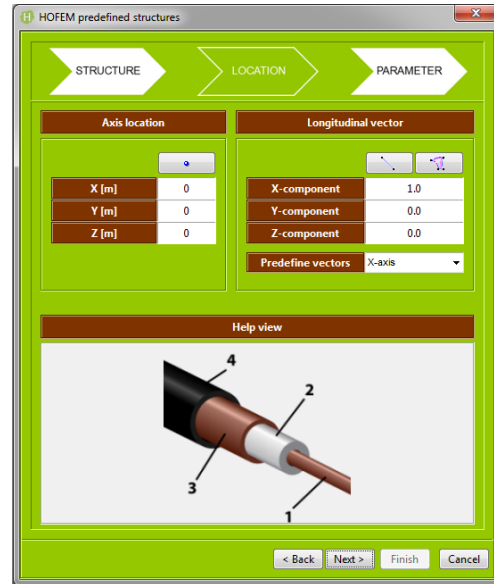
El cable coaxial otro de los medios más utilizados para proporcionar la excitación en problemas de alta frecuencia. Es una estructura algo más compleja que la guía rectangular, pero igualmente de uso extendido. La Fig. 2.10 muestra el siguiente paso del asistente para el cable coaxial. Está dividido en dos áreas, en la parte superior, solicita los primeros parámetros necesarios y otra en la parte inferior, denominada 'Help view', donde se representa a modo de ayuda una imagen detallada de un cable coaxial.

Para poder generar el cable coaxial, el asistente solicita en primer lugar el punto de origen desde el que se empezará a dibujar, denominado 'Axis Location', se introducirá de este modo las coordenadas X,Y,Z del punto de origen (véase la Fig. 2.11) También existe la posibilidad de seleccionar un punto ya existente de una estructura anterior, pulsando el botón 'Pick' . A continuación, hay que introducir el vector que definirá el eje longitudinal del cable coaxial. Para ello se solicitan las coordenadas «X,Y,Z» de dicho vector. El asistente cuenta con un desplegable en el que poder seleccionar uno de los vectores predefinidos, que serán «X,Y,Z,-X,-Y,-Z» (véase la Fig. 2.12). Adicionalmente, pinchando en el botón 'Pick'  se puede seleccionar un vector ya existente de una estructura anterior.

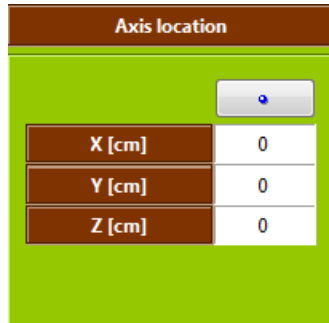
## 2.3. DEFINICIÓN DE LOS PASOS DEL ASISTENTE



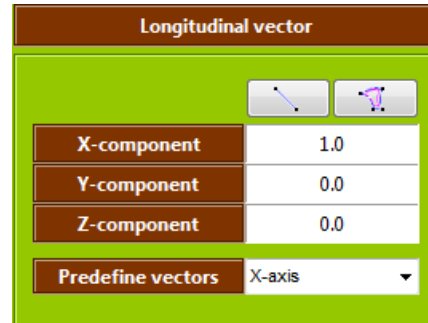
**Figura 2.9:** Ventana de selección de estructuras



**Figura 2.10:** Localización de la estructura



**Figura 2.11:** Localización del Eje.



**Figura 2.12:** Vector Longitudinal.

Una vez introducidos los parámetros requeridos, es necesario pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso. A la izquierda de este botón, se encuentra otro denominado «Back» con el que regresar al paso anterior para seleccionar una estructura diferente. Si por el contrario se desea abandonar el asistente, es necesario pulsar el botón «Cancel» situado en la parte inferior derecha.

### 2.3.5 Cable Coaxial: Parámetros

La Fig. 2.13 muestra el último paso del asistente para el cable coaxial. En este punto se solicitan los parámetros básicos de un cable coaxial, como son los radios interior y exterior, representadas por 'Parameter (R1)', 'Parameter (R2)' y las longitudes del cable completo y la del 'vivo', representadas por 'Parameter (L1)' y 'Parameter (L2)'.



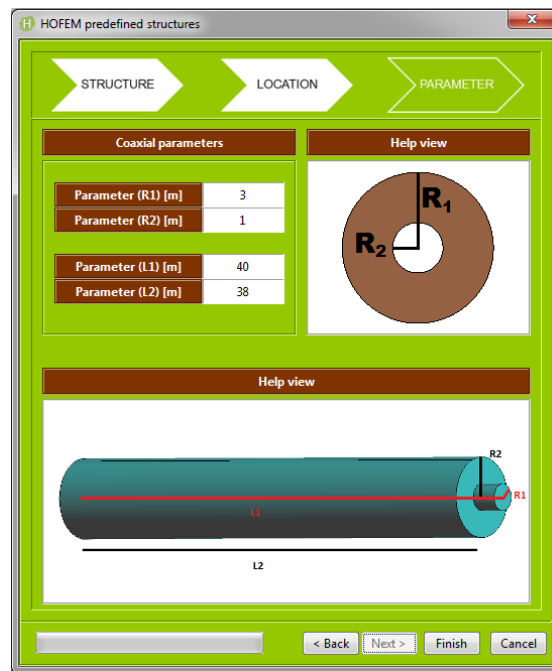


Figura 2.13: Parámetros.

Se dispone de dos ventanas de ayuda, representadas como «Help View» que muestran a qué corresponden cada uno de los parámetros solicitados. La primera de ellas mostrará una imagen de un corte del cable coaxial, la segunda muestra una vista lateral del mismo. Una vez introducidos todos los parámetros, pulsando en el botón «Finish», se procede a la creación del cable coaxial conforme a los datos que se han ido introduciendo.

### 2.3.6 Cable Coaxial: Creación

Hasta este punto es donde el usuario puede ir interactuando con el asistente, eligiendo como quiere construir el cable coaxial conforme a los parámetros que se han ido introduciendo. A partir de aquí se va a exponer como se lleva a cabo la creación de la estructura.

Tras pulsar el botón «Finish» se invoca el proceso 'CoaxialCable'. En este proceso, en primer lugar, se realiza una normalización del vector que conforma el eje longitudinal del cable coaxial (véanse líneas 3-5 del Pseudocódigo 2.17). A continuación, se definen los valores de los radios y las longitudes del cable coaxial (véanse líneas 7-12 del Pseudocódigo 2.17). Posteriormente se definen los puntos relevantes para la construcción del cable coaxial. Se establece el centro desde el cual se comienza a dibujar la estructura (líneas 3-5 del Pseudocódigo 2.18), el centro sobre el que dibujar la camisa del cable coaxial (líneas 9-11 del Pseudocódigo 2.18) y el centro sobre el que dibujar la tapa del cable en el caso de los problemas de radiación (líneas 15-17 del Pseudocódigo 2.18).

### 2.3. DEFINICIÓN DE LOS PASOS DEL ASISTENTE

```
1 proc CoaxialCable { } {
2
3 #Normalización del vector que conforma el eje longitudinal
4   set nn [MathUtils::VectorNormalized [list $vect(x)
5                                         $vect(y) $vect(z)]]
6
7 #Definición de las dimensiones del cable coaxial
8   set radio1 $param(rr)
9   set radio2 $param(rr2)
10  set punta $param(ll)
11  set largo1 [expr $param(ll)+$param(ll2)]
12  set largo2 $param(ll2)
13  ...
14 }
```

**Pseudocódigo 2.17:** Definir Parámetros

```
1 #Definición del punto origen sobre el que se dibuja el vivo
2 #   del cable Coaxial
3   set cent_x $cent(x)
4   set cent_y $cent(y)
5   set cent_z $cent(z)
6 #Definición del punto origen sobre el que se dibuja la
7 #camisa del cable
8   set cent2_x [expr $cent_x + $punta*[lindex $nn 0]]
9   set cent2_y [expr $cent_y + $punta*[lindex $nn 1]]
10  set cent2_z [expr $cent_z + $punta*[lindex $nn 2]]
11 #Definición del punto origen sobre el que se dibuja la tapa
12 #   del cable coaxial
13   set cent3_x [expr $cent_x + 0.07*$largo2*[lindex $nn 0]]
14   set cent3_y [expr $cent_y + 0.07*$largo2*[lindex $nn 1]]
15   set cent3_z [expr $cent_z + 0.07*$largo2*[lindex $nn 2]]
```

**Pseudocódigo 2.18:** Definir Centros

A continuación se procede a dibujar el cilindro correspondiente al interior del cable coaxial o «vivo» (ver líneas 2-5 del Pseudocódigo 2.19). Por otro lado se genera un nuevo cilindro que corresponderá a la camisa del cable (líneas 7-9 del Pseudocódigo 2.19). Una vez creados estos dos cilindros, se procede a eliminar los volúmenes creados (véanse líneas 2-3 del Pseudocódigo 2.20) así como las superficies de las bases (líneas 6-8 del Pseudocódigo 2.20). A continuación se crean las nuevas bases en forma de anillo (líneas 11-17 del Pseudocódigo 2.20) para posteriormente generar el volumen que comprende los dos cilindros concéntricos (véanse líneas 20-24 del Pseudocódigo 2.20).

### 2.3. DEFINICIÓN DE LOS PASOS DEL ASISTENTE

```
1 #Dibujar el cilindro interior
2   GiD_Process Geometry Create Object Cylinder
3     $cent2_x $cent2_y $cent2_z
4     $vect(x) $vect(y) $vect(z) $radio2 $largo2
5   GiD_Process Mescape
6 #Dibujar el cilindro exterior
7   GiD_Process Geometry Create Object Cylinder
8     $cent2_x $cent2_y $cent2_z
9     $vect(x) $vect(y) $vect(z) $radio1 $largo2
10  GiD_Process Mescape
```

**Pseudocódigo 2.19:** Origen

```
1 #Eliminar los volúmenes existentes
2   GiD_Process Geometry Delete Volumes
3     [expr $volumenes+1] [expr $volumenes+2]
4 #Eliminar las superficies de las bases
5   GiD_Process Geometry Delete Surfaces
6     [expr $superficies+3] [expr $superficies+4]
7     [expr $superficies+7] [expr $superficies+8]
8 #Crear las nuevas superficies de las bases
9   GiD_Process Geometry create NurbsSurface
10     [expr $lineas+3] [expr $lineas+5]
11     [expr $lineas+9] [expr $lineas+11]
12   GiD_Process Geometry create NurbsSurface
13     [expr $lineas+4] [expr $lineas+6]
14     [expr $lineas+10] [expr $lineas+12]
15 #Definir el nuevo volumen
16   GiD_Process Geometry create Volume
17     [expr $superficies+1] [expr $superficies+2]
18     [expr $superficies+5] [expr $superficies+6]
19     [expr $superficies+7] [expr $superficies+8]
20 ...
```

**Pseudocódigo 2.20:** Eliminar Contornos

En este momento ya se tiene el volumen con los dos cilindros concéntricos perfectamente definidos, faltaría por crear el cilindro correspondiente a la punta saliente del vivo. Para ello se realiza una traslación de la circunferencia de la base del cilindro hasta el punto origen definido al inicio (véanse líneas 2-5 del Pseudocódigo 2.21). En el caso de los problemas de radiación, se realiza una traslación de la circunferencia de radio mayor hasta el lugar donde irá la tapa (véanse líneas 7-10 del Pseudocódigo 2.21). Por último se crea la superficie donde irá el puerto (véanse líneas 12-15 del Pseudocódigo 2.21)

```
1 #Crear el cilindro correspondiente a la punta del vivo
2 GiD_Process Utilities Copy Lines DoExtrude Surfaces
3     Translation $cent2_x $cent2_y $cent2_z $cent_x
4     $cent_y $cent_z [expr $lineas+3] [expr $lineas+5]
5 GiD_Process Mescape
6 #Crear el cilindro correspondiente a la tapa del coaxial
7 GiD_Process Utilities Copy Lines DoExtrude Surfaces
8     Translation $cent_x $cent_y $cent_z $cent3_x
9     $cent3_y $cent3_z [expr $lineas+12] [expr $lineas+10]
10 GiD_Process Mescape
11 #Crear la superficie del puerto
12 GiD_Process Geometry create NurbsSurface
13     [expr $lineas+17] [expr $lineas+18]
14 GiD_Process Mescape
15     ...
```

**Pseudocódigo 2.21:** Dibujar superficies

## 2.4 Conclusiones

En este capítulo hemos visto como desarrollar un asistente que nos permita el modelado de algunas estructuras. Para ello se ha utilizado el software de propósito general GID como base y mediante la codificación de algunos archivos usando el mismo lenguaje en el que está escrito GID, hemos conseguido implementar el asistente que permite generar estas estructuras únicamente introduciendo sus parámetros característicos.

En la primera parte se ha mostrado como implementar este asistente, explicando en primer lugar los métodos que se encuentran ya predefinidos en los paquetes que se han utilizado y, continuando con los métodos diseñados por nosotros para dotar a la herramienta de la funcionalidad deseada. También se muestra como dotar al asistente de la apariencia elegida. En una segunda parte, se explica como utilizar el asistente creado desde como se invoca, hasta como introducir los parámetros necesarios según el tipo de estructura escogida. Por último, también se indica de forma detallada, como se construyen las estructuras seleccionadas en base a elementos primarios.

Este asistente está preparado para que se puedan añadir ciertas mejoras. Una de estas mejoras puede ser dotar de funcionalidad al botón «pick» para seleccionar el vector de posicionamiento en base a uno ya existente. También se pueden incluir nuevas estructuras de forma sencilla, como pueden ser antenas planares. En una fase inicial y no depurada del asistente, se incluían más estructuras pero finalmente se decidió no incluirlas ya que no iban a ser utilizadas en las secciones posteriores del proyecto y este asistente pretende ser una muestra de las posibilidades que ofrece GID a la hora de su customización y mostrar las ventajas en término de ahorro de tiempo de cara a su utilización posterior.

## 2.4. CONCLUSIONES

---

La forma de utilizar el asistente desarrollado y un ejemplo directo de su aplicación a la hora de diseñar estructuras complejas, se muestra en el Anexo A. Como se puede ver en cualquiera de los ejemplos del Anexo A, la introducción de este asistente en la herramienta GID, permite la definición de los puertos de excitación de una forma mucho más sencilla, lo que facilita sensiblemente la tarea al usuario respecto a realizarlo de forma manual, reduciendo de este modo el tiempo destinado al modelado de estructuras.



---

---

# CAPÍTULO 3

---

## VERIFICACIÓN DE RESULTADOS

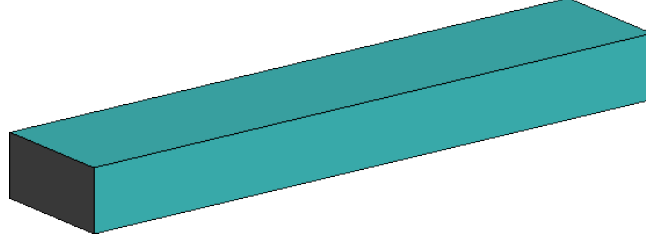
### 3.1 Introducción

Tal y como se ha comentado en el Capítulo 1, una de las tareas más importantes una vez terminado el desarrollo de un código de simulación es la verificación de sus resultados. Cualquier código de simulación, antes de poder ser utilizado, ya sea de manera comercial o interna dentro de un grupo de investigación, debe pasar por unas pruebas o *benchmarks* para comprobar su correcto funcionamiento. Típicamente estas pruebas se basan en la simulación de problemas de los que se conocen sus resultados analíticos de antemano, y por tanto, basta con comparar estos resultados con los dados por el código para saber si los resultados son correctos.

Este capítulo se centra en la verificación de los resultados proporcionados por el código de Elementos Finitos descrito en la introducción de este PFC. Como se trata de un código electromagnético de propósito general las pruebas que se deben realizar han de ser completas en términos de tipos de problemas a simular. De este modo, las pruebas para la verificación del código se han dividido en problemas de guías de ondas y problemas de radiación.

### 3.2 Problemas de guías de ondas

La primera batería de pruebas va a servir para comprobar que las partes básicas del código, tales como la asignación de condiciones de contorno o la excitación de las guías, funcionan correctamente. Las pruebas irán aumentando su dificultad para observar también la respuesta del código en cuanto a rendimiento. Al finalizar esta batería de pruebas se expondrán las conclusiones alcanzadas para este grupo de pruebas.



**Figura 3.1:** Geometría de la guía rectangular utilizada en la primera prueba

### 3.2.1 Prueba 1: Guía rectangular

La primera estructura que se va a analizar es una guía rectangular. Esta estructura ha sido seleccionada para realizar una primera verificación del código, especialmente para probar las condiciones de contorno correspondientes a la excitación del problema. La geometría de esta estructura se muestra en la Fig. 3.1. La guía rectangular es una estructura que no presenta discontinuidades a lo largo de su eje longitudinal, por lo que la solución del campo es conocida analíticamente. De esta forma, los valores analíticos de los parámetros  $S$  son  $S_{11} = S_{22} = 0$  y  $S_{21} = S_{12} = e^{-j\beta_{10}l}$ , donde  $l$  es la longitud de la guía.

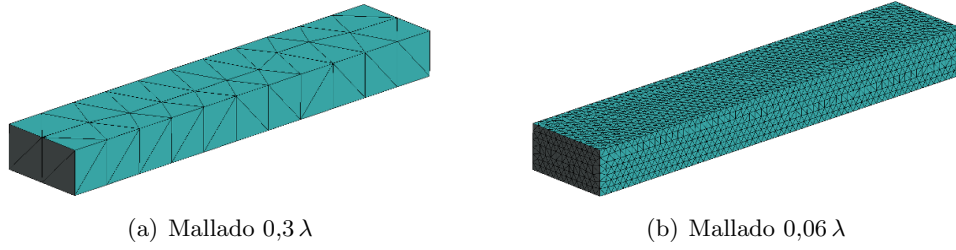
Las dimensiones de la guía considerada para la verificación del código son tales que la relación entre el ancho de la guía  $a$  y el alto  $b$  es  $\frac{a}{b} = 2$ . La longitud de la guía se ha considerado que mide  $l = 5a$ . Las dimensiones exactas de la guía utilizada en esta verificación son  $a = 4$  cm,  $b = 2$  cm y  $l = 20$  cm. Utilizando cualquier guía que cumpla estas dimensiones se obtienen los resultados mostrados en la Tabla 3.1. Estos resultados se han tomando para una valor de frecuencia de  $K = 1.5 K_c$ , cuyo valor es de 5.625 GHz para las dimensiones exactas descritas anteriormente. Además, para ver la importancia del mallado en los resultados, se han generado diversos mallados con tamaños que van desde  $0,3\lambda$  a  $0,06\lambda$ . Véanse las Fig. 3.2(a) y 3.2(b) que muestran los mallados para los tamaños mencionados anteriormente.

**Tabla 3.1:** Parámetros S de la Guía Rectangular

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$  |
|---------------|----------------|------------|------------|------------|-----------------|
| $0.3\lambda$  | 235            | 2.35       | 5.912E-2   | 0.9897006  | $-113.19^\circ$ |
| $0.2\lambda$  | 896            | 2.80       | 3.513E-3   | 0.9996744  | $-105.97^\circ$ |
| $0.1\lambda$  | 10987          | 16.56      | 6.567E-5   | 0.9999876  | $-106.21^\circ$ |
| $0.09\lambda$ | 15906          | 29.89      | 8.591E-6   | 0.9999903  | $-106.22^\circ$ |
| $0.08\lambda$ | 23785          | 52.21      | 4.873E-6   | 0.9999944  | $-106.22^\circ$ |
| $0.07\lambda$ | 36853          | 120.95     | 4.529E-6   | 0.9999973  | $-106.22^\circ$ |
| $0.06\lambda$ | 61590          | 318.96     | 3.719E-6   | 0.9999984  | $-106.22^\circ$ |
| Analítico     | —              | —          | 0.0        | 1.0        | $-106.24^\circ$ |



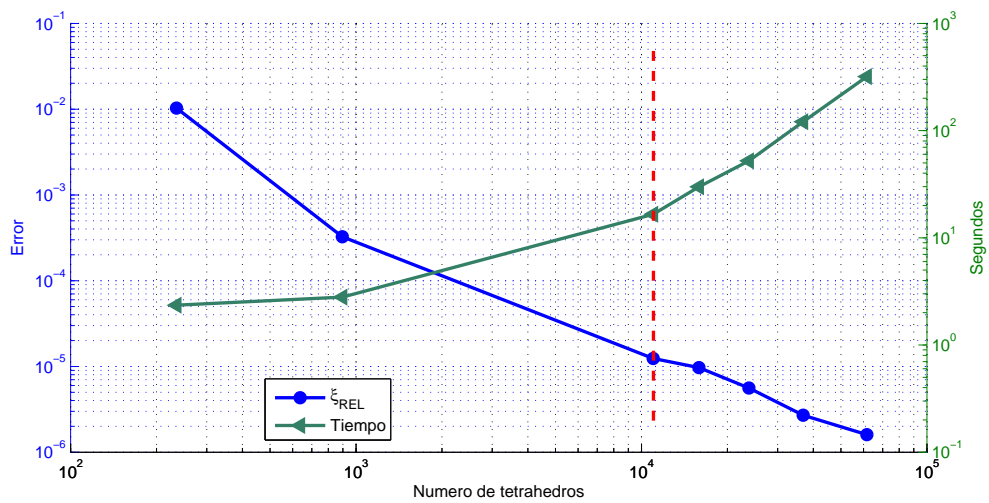
### 3.2. PROBLEMAS DE GUÍAS DE ONDAS



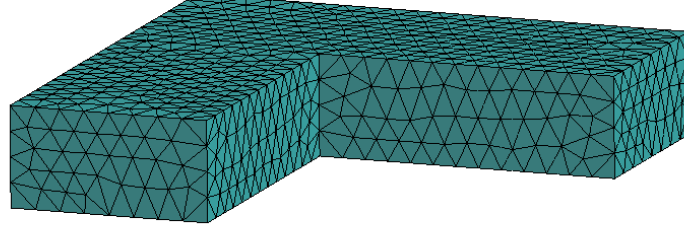
**Figura 3.2:** Mallados de la guía rectangular

A la vista de los resultados, se comprueba que el error va disminuyendo conforme aumenta la discretización del mallado. A partir de  $0.2 \lambda$  se obtiene un valor con un error en torno al 0.1 %. La mejora en los resultados al mallar con un tamaño de  $0.09\lambda$  con respecto a  $0.1\lambda$ , no es sustancial, y no compensa el incremento de número de elementos y del tiempo de cómputo. Por lo que para simular esta estructura, un mallado entorno a  $0.1 \lambda$  sería suficiente para obtener unos resultados muy buenos con un error por debajo del 0.1 %

La Fig. 3.3 muestra la convergencia del error del parámetro  $|S_{21}|$  conforme aumenta la discretización del mallado. Paralelamente a esto, muestra el tiempo de cómputo a medida que aumenta dicho mallado. A partir de  $0.1 \lambda$  (véase la línea roja vertical), la pendiente del tiempo cambia su tendencia mientras que la del error permanece constante. De esta forma, se puede concluir que este tamaño de mallado presenta un buen compromiso entre el error obtenido y el tiempo de cómputo.



**Figura 3.3:** Convergencia del error y tiempo de cómputo para una guía rectangular.



**Figura 3.4:** Codo Plano H de  $90^\circ$ .

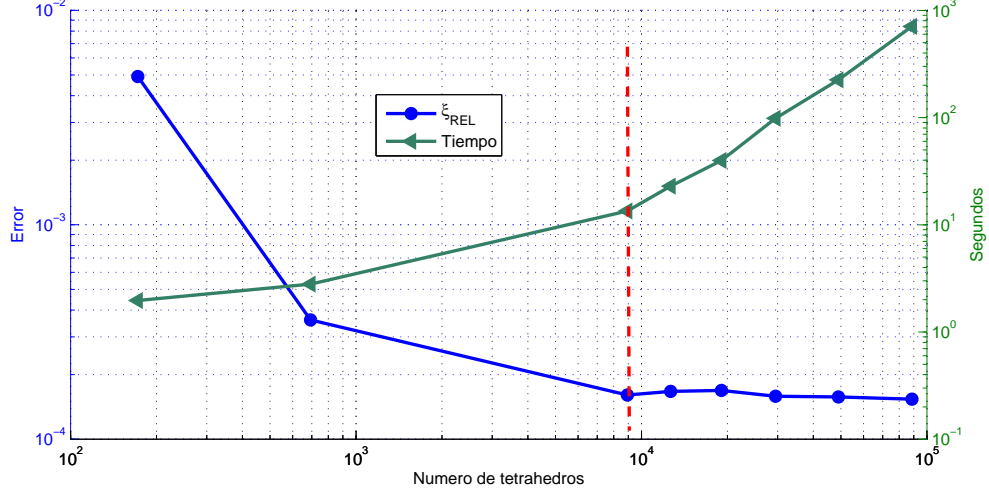
### 3.2.2 Prueba 2: Codo plano-H de $90^\circ$

La segunda estructura bajo análisis es un codo plano-H de  $90^\circ$ . Esta estructura, como en el caso de la guía rectangular, es recíproca por lo que  $S_{11} = S_{22}$  y  $S_{21} = S_{12}$ . El codo considerado presenta unas dimensiones tales que la relación entre el ancho de la guía  $a$  y el alto  $b$  sea  $\frac{a}{b} = 2$ . Además, las dimensiones de ambos extremos son iguales de longitud  $l_1 = l_2 = 1.5a$ . En particular, las medidas seleccionadas son  $a = 4$  cm,  $b = 2$  cm,  $l_1 = 6$  cm y  $l_2 = 6$  cm. Un ejemplo de codo plano H se muestra en la Fig. 3.4.

En este segundo ejemplo, los valores con lo que comparar la solución del código han sido obtenidos mediante la técnica semianalítica de Mode Matching (MM) [18, 19, 20]. Estos resultados se muestran en la Tabla 3.2, donde cualquier codo plano-H que tenga las dimensiones anteriormente citadas presenta dichos parámetros  $S$ . Los resultados han sido obtenidos para una frecuencia de 5.625 GHz. Además, para ver la importancia del mallado en los resultados, se han generado diversos mallados con tamaños que van desde  $0,3\lambda$  a  $0,05\lambda$ .

**Tabla 3.2:** Parámetros S del Codo Plano H de  $90^\circ$

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$  |
|---------------|----------------|------------|------------|------------|-----------------|
| $0.3\lambda$  | 172            | 1.97       | 0.4125981  | 0.9050047  | $-66.93^\circ$  |
| $0.2\lambda$  | 693            | 2.80       | 0.4155154  | 0.9091480  | $-62.85^\circ$  |
| $0.1\lambda$  | 8926           | 13.44      | 0.4154178  | 0.9096219  | $-63.09^\circ$  |
| $0.09\lambda$ | 12664          | 22.95      | 0.4154076  | 0.9096275  | $-63.10^\circ$  |
| $0.08\lambda$ | 19073          | 39.79      | 0.4154120  | 0.9096291  | $-63.11^\circ$  |
| $0.07\lambda$ | 29509          | 98.39      | 0.4154366  | 0.9096199  | $-63.11^\circ$  |
| $0.06\lambda$ | 49007          | 224.66     | 0.4154408  | 0.9096188  | $-63.12^\circ$  |
| $0.05\lambda$ | 88615          | 708.22     | 0.4154494  | 0.9096155  | $-63.12^\circ$  |
| (MM)          | —              | —          | 0.4155423  | 0.9094756  | $-63.133^\circ$ |



**Figura 3.5:** Convergencia del error conforme aumenta el mallado para un codo plano H.

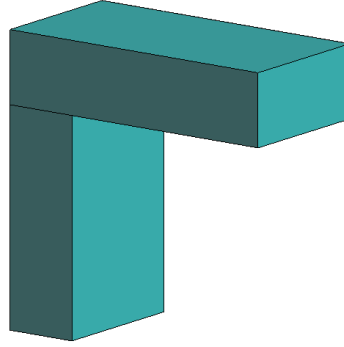
La Fig 3.5 la convergencia del error según aumenta el número de incógnitas del problema. Se puede ver como a partir de  $0.1 \lambda$  (línea discontinua de color rojo en la figura) el error permanece prácticamente constante, mientras que el tiempo de ejecución aumenta de forma drástica, por lo que se puede concluir que este tamaño resulta un buen tamaño para realizar el cálculo de los parámetros S.

### 3.2.3 Prueba 3: Codo plano-E de $90^\circ$

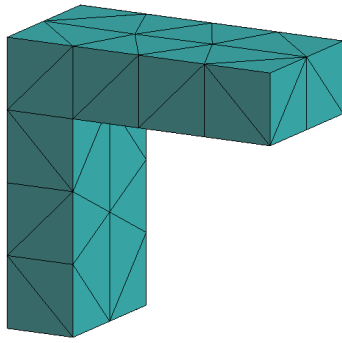
La tercera estructura bajo análisis es un codo plano-E de  $90^\circ$  (ver Fig.3.6). Es dual al codo plano-H de  $90^\circ$ , y al igual que la anterior, es muy utilizada en los circuitos de microondas. Las dimensiones son las mismas al codo plano-H de  $90^\circ$ , por lo tanto se cumple que la relación entre el ancho  $a$  y el alto de la guía  $b$  es tal que  $\frac{a}{b} = 2$ . Además, debe cumplirse que  $l_1 = l_2 = 1.5a$ . Al igual que en los dos casos anterior  $S_{11} = S_{22}$  y  $S_{21} = S_{12}$ . La diferencia de estas dos estructuras reside en que en el plano-H, la curvatura tiene lugar por el plano horizontal, y en el caso del plano-E la curvatura se produce por el plano vertical. En este caso en particular, las medidas seleccionadas son  $a = 4$  cm,  $b = 2$  cm,  $l_1 = 6$  cm y  $l_2 = 6$  cm.

Los resultados de la simulación se han comparado con los obtenidos mediante MM y se muestran en la Tabla 3.3. Estos resultados se han tomando para un valor de frecuencia de 5.625 GHz. Los mallados para  $0.3 \lambda$  y  $0.05 \lambda$  se muestran en las Fig. 3.7(a) y 3.7(b). La Fig. 3.8 muestra la evolución del error y el tiempo de cómputo frente a la discretización del mallado. En este caso, desde la primera simulación se obtienen unos valores próximos a los teóricos. De hecho, a  $0.1 \lambda$  se observa un ligero aumento en el mismo, rondando valores algo superiores al 1%. A partir de un tamaño en el mallado de  $0.09 \lambda$  el error apenas disminuye, mientras que el tiempo de cómputo continúa creciendo, por lo que elegiremos este tamaño para calcular los parámetros. La línea roja discontinua de la figura indica los datos para el mallado de  $0.09 \lambda$ .

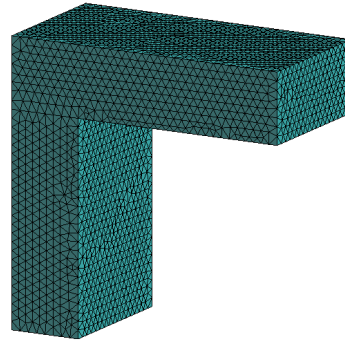
### 3.2. PROBLEMAS DE GUÍAS DE ONDAS



**Figura 3.6:** Codo Plano-E de  $90^\circ$ .



(a) Mallado  $0.3 \lambda$

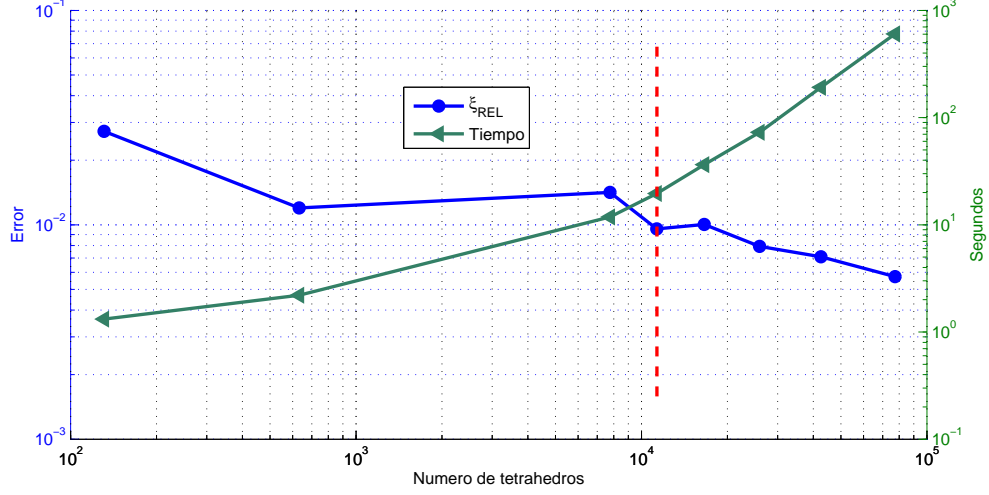


(b) Mallado  $0.05 \lambda$

**Figura 3.7:** Mallados Codo plano E

**Tabla 3.3:** Parámetros S del Codo Plano E de  $90^\circ$

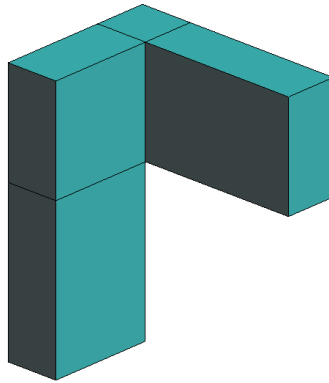
| Tamaño         | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$ |
|----------------|----------------|------------|------------|------------|----------------|
| $0.3 \lambda$  | 131            | 1.32       | 0.5366067  | 0.8287690  | $15.75^\circ$  |
| $0.2 \lambda$  | 632            | 2.20       | 0.5450387  | 0.8344712  | $19.00^\circ$  |
| $0.1 \lambda$  | 7761           | 11.84      | 0.5438393  | 0.8391772  | $19.80^\circ$  |
| $0.09 \lambda$ | 11313          | 19.61      | 0.5463743  | 0.8375314  | $19.83^\circ$  |
| $0.08 \lambda$ | 16593          | 36.45      | 0.5461171  | 0.8377036  | $19.97^\circ$  |
| $0.07 \lambda$ | 25933          | 72.98      | 0.5472740  | 0.8369507  | $20.05^\circ$  |
| $0.06 \lambda$ | 42540          | 191.69     | 0.5477368  | 0.8366491  | $20.09^\circ$  |
| $0.05 \lambda$ | 77306          | 603.47     | 0.5484899  | 0.8361563  | $20.14^\circ$  |
| (MM)           | —              | —          | 0.551652   | 0.8341653  | $20.29^\circ$  |



**Figura 3.8:** Convergencia del error conforme aumenta el mallado para un codo plano E.

#### 3.2.4 Prueba 4: Codo plano-H + plano-E de 90°

La siguiente estructura bajo análisis es la composición o cascada de un codo plano-H y un codo plano-E de 90° (véase la Fig. 3.9). Debido a la interacción entre los modos superiores en las dos discontinuidades, no podemos calcular los parámetros S de la cascada como una superposición de las dos estructuras por separado. Por este motivo se van a analizar dos casos diferentes con distinta separación entre el codo plano-H y el codo plano-E. Las dimensiones cumplen que el ancho de la guía  $a$  y el alto  $b$  es tal que  $\frac{a}{b} = 2$ , además  $l_1 = l_2 = 1.5a$ , y en el primer caso la separación es  $d = 0$ . En este caso en particular, las medidas seleccionadas son  $a = 4$  cm,  $b = 2$  cm,  $l_1 = 6$  cm y  $l_2 = 6$  cm, para cuyos valores la frecuencia de corte del modo  $TE_{10}$  es  $K_{c10} = 3.75$  GHz. Los resultados de las simulaciones se han tomado para un valor de frecuencia de  $K = 1.5K_c$  que, para este caso en particular, es de 5.625 GHz.



**Figura 3.9:** Cascada de codo plano-H y plano-E de 90° con  $d = 0$

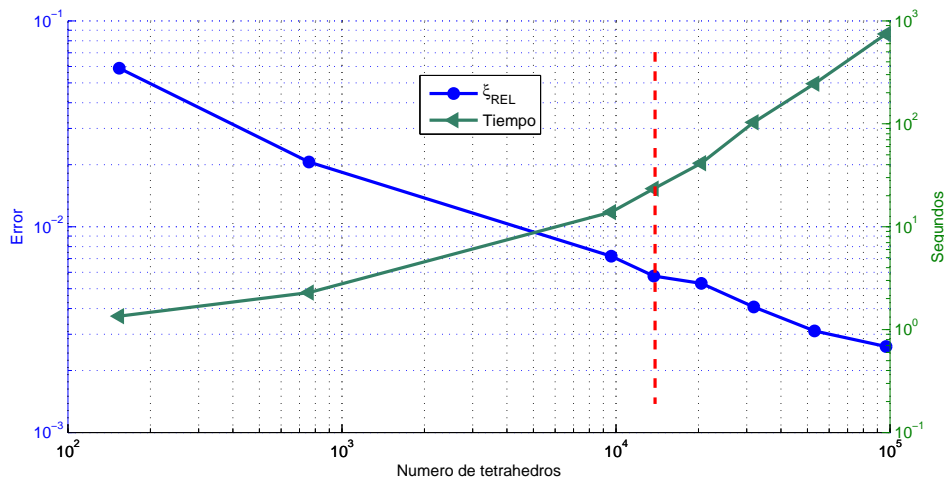
### 3.2. PROBLEMAS DE GUÍAS DE ONDAS

**Tabla 3.4:** Parámetros S de la cascada de codo Plano-H y codo Plano-E de  $90^\circ$  con  $d = 0$

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$ |
|---------------|----------------|------------|------------|------------|----------------|
| $0.3\lambda$  | 154            | 1.35       | 0.6916661  | 0.7112671  | $27.58^\circ$  |
| $0.2\lambda$  | 758            | 2.29       | 0.6357882  | 0.7713442  | $31.98^\circ$  |
| $0.1\lambda$  | 9609           | 13.87      | 0.6485103  | 0.7611885  | $32.73^\circ$  |
| $0.09\lambda$ | 13744          | 23.45      | 0.6497873  | 0.7601027  | $32.84^\circ$  |
| $0.08\lambda$ | 20476          | 41.37      | 0.6501942  | 0.7597608  | $32.85^\circ$  |
| $0.07\lambda$ | 31847          | 103.45     | 0.6512804  | 0.7588331  | $32.95^\circ$  |
| $0.06\lambda$ | 53022          | 244.91     | 0.6521261  | 0.7581081  | $33.03^\circ$  |
| $0.05\lambda$ | 96768          | 746.30     | 0.6525661  | 0.7577305  | $33.05^\circ$  |
| (MM)          | —              | —          | 0.6550234  | 0.7557532  | $33.25^\circ$  |

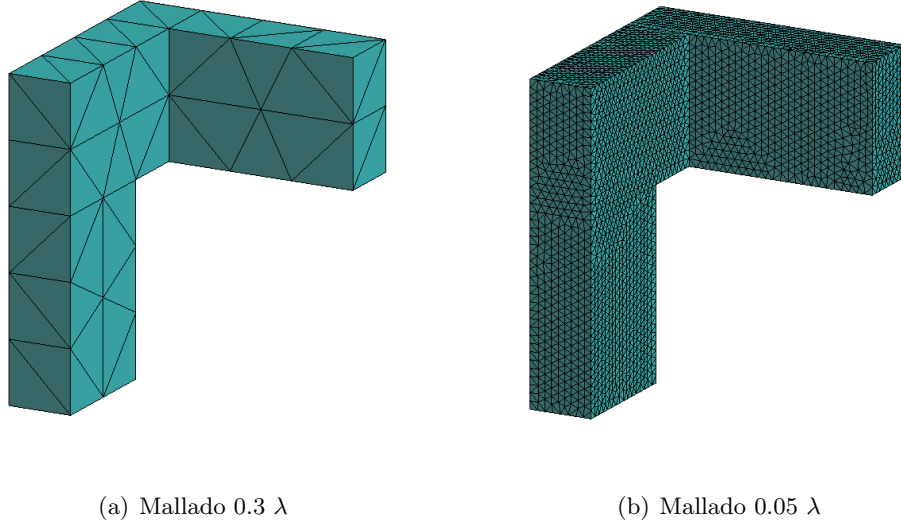
Los resultados de la simulación, y su comparación con los dados por la técnica de MM, se muestran en la Tabla 3.4. La Fig. 3.10 muestra como el error va disminuyendo a medida que aumenta el mallado. Esta reducción del error es prácticamente lineal, mientras que el tiempo de cómputo aumenta considerablemente a partir de  $0.1\lambda$ . En este punto se obtiene un error en torno al 1.0 %. Para obtener un error inferior al 1.0 % el mallado debe ser inferior a  $0.1\lambda$ . Para simular esta estructura, un mallado de  $0.09\lambda$  (línea roja discontinua en la figura) es más que suficiente para obtener resultados precisos sin que el tiempo de cómputo o el número de elementos sean elevados.

A continuación se analiza el caso en el que se tiene una separación de  $d = 0.5a$ . Los mallados de esta estructura se muestran en las Fig. 3.11(a) y 3.11(b). Las medidas utilizadas en este caso son  $a = 4$  cm,  $b = 2$  cm,  $l_1 = 6$  cm,  $l_2 = 6$  cm y  $d = 2$  cm.



**Figura 3.10:** Convergencia del error para un codo plano EH con  $d = 0$ .

### 3.2. PROBLEMAS DE GUÍAS DE ONDAS

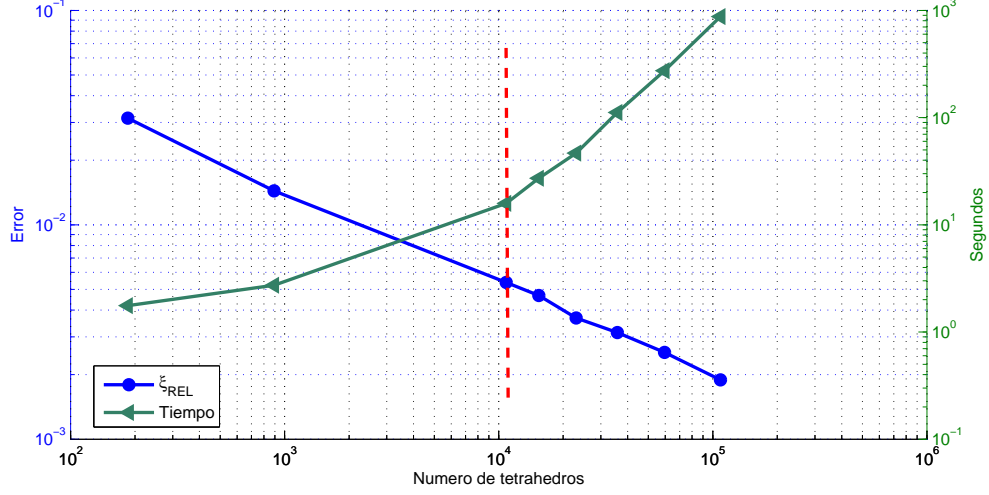


**Figura 3.11:** Mallados codo plano H + codo plano E

Al igual que en las pruebas anteriores los resultados de las simulaciones se han tomando para un valor de frecuencia de 5.625 GHz. La Tabla 3.5 muestra la comparación de los resultados dados por el código y los obtenidos mediante la técnica de MM. La Fig. 3.12 muestra como el error va disminuyendo a medida que aumenta el mallado. El error va decreciendo de forma lineal en todas las simulaciones, mientras que el tiempo de cómputo comienza a aumentar en mayor medida a partir de  $0.1 \lambda$ . En este punto se obtiene un error en torno al 0.5 %. La línea roja discontinua de la figura muestra la opción de mallado elegida. En este caso, un mallado de  $0.1 \lambda$  es más que suficiente para obtener resultados precisos con un tiempo de ejecución reducido.

**Tabla 3.5:** Parámetros S de la cascada de codo plano-H y plano-E de  $90^\circ$  con  $d = 0.5 a$

| Tamaño         | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$ |
|----------------|----------------|------------|------------|------------|----------------|
| $0.3 \lambda$  | 185            | 1.76       | 0.7444070  | 0.6558238  | $-99.70^\circ$ |
| $0.2 \lambda$  | 892            | 2.73       | 0.7264198  | 0.6868633  | $-90.80^\circ$ |
| $0.1 \lambda$  | 10842          | 15.87      | 0.7324996  | 0.6807571  | $-90.51^\circ$ |
| $0.09 \lambda$ | 15389          | 27.21      | 0.7329413  | 0.6802830  | $-90.48^\circ$ |
| $0.08 \lambda$ | 22983          | 46.59      | 0.7335780  | 0.6796005  | $-90.42^\circ$ |
| $0.07 \lambda$ | 35823          | 111.51     | 0.7339139  | 0.6792399  | $-90.39^\circ$ |
| $0.06 \lambda$ | 59484          | 274.21     | 0.7338053  | 0.6788322  | $-90.32^\circ$ |
| $0.05 \lambda$ | 108556         | 876.62     | 0.7346989  | 0.6783926  | $-90.33^\circ$ |
| (MM)           | —              | —          | 0.7359231  | 0.6771123  | $-90.23^\circ$ |

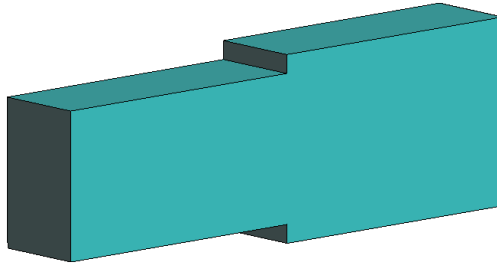


**Figura 3.12:** Convergencia del error para un codo plano EH con  $d = 0.5a$ .

### 3.2.5 Prueba 5: Cambio de ancho

La siguiente estructura analizada es una guía rectangular que tiene un cambio en el ancho de la guía, tal como se puede ver en la Fig. 3.13. Tiene una geometría en forma de L en el plano H que permanece invariante en la dirección del borde estrecho de la guía. La variación del campo a lo largo de la dirección perpendicular al plano de la discontinuidad es nula. Las dimensiones son tales que el ancho de la guía  $a$  y el alto  $b$  cumplen que  $\frac{a_1}{b} = 2$ ,  $a_2 = 0,8a_1$  y además las longitudes son  $l_1 = l_2 = 3b$ . En este caso en particular, las medidas seleccionadas son  $b = 2$  cm,  $a_1 = 4$  cm,  $a_2 = 3,2$  cm,  $l_1 = 6$  cm y  $l_2 = 6$  cm.

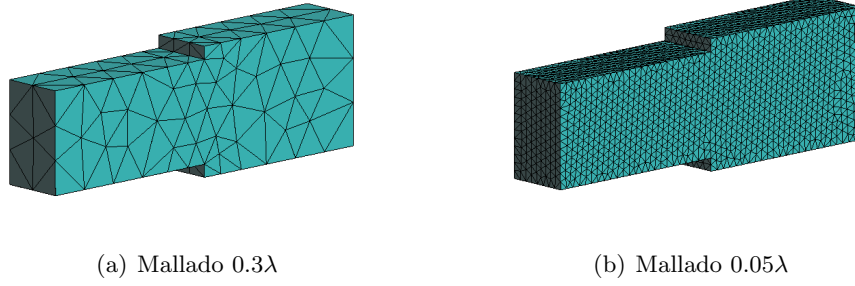
Siguiendo el mismo esquema que en las pruebas realizadas anteriormente, los resultados de las simulaciones se han tomado para un valor de frecuencia de 5.625 GHz. Además, se han vuelto a generar diversos mallados con tamaños que van desde  $0,3\lambda$  a  $0,05\lambda$  como los mostrados en las Fig. 3.14(a) y 3.14(b).



**Figura 3.13:** Cambio de ancho de la guía



### 3.2. PROBLEMAS DE GUÍAS DE ONDAS



**Figura 3.14:** Cambio de ancho de la guía

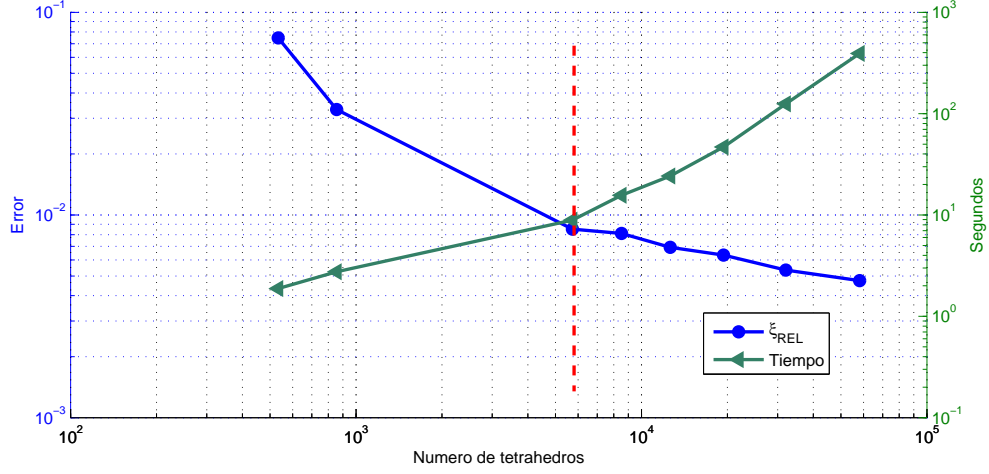
Los resultados de la prueba se muestran en la Tabla 3.6. La Fig. 3.15 muestra la convergencia del error a medida que la discretización del mallado aumenta. Para los mallados de  $0.3$  y  $0.2\lambda$  el error obtenido es muy grande, a partir de  $0.1\lambda$ , el error se encuentra por debajo del  $1\%$ . A partir de aquí la reducción en el error es mínima, mientras que el tiempo de cómputo empieza a crecer de forma significativa. Por lo que para simular esta estructura podríamos elegir un tamaño en el mallado de  $0.1\lambda$ .

#### 3.2.6 Prueba 6: Cambio del alto

La siguiente estructura es una guía que tiene un cambio en la altura de la guía, tal como se puede ver en las Fig. 3.16(a) y 3.16(b). Tiene una geometría en forma de L en el plano E que permanece invariante en el ancho de la guía. Al contrario de lo que sucede en el caso anterior, la variación del campo a lo largo de la dirección perpendicular al plano de la discontinuidad no es nula. Las dimensiones son tales que el ancho de la guía  $a$  y el alto  $b$  cumplen que  $\frac{a}{b_1} = 2$ ,  $\frac{b_1}{2} = b_2$  además  $l_1 = l_2 = 2a$ . En este caso en particular, las medidas seleccionadas son  $a = 4$  cm,  $b_1 = 2$  cm,  $b_2 = 1$  cm,  $l_1 = 8$  cm y  $l_2 = 8$  cm.

**Tabla 3.6:** Parámetros S del cambio de ancho de la guía

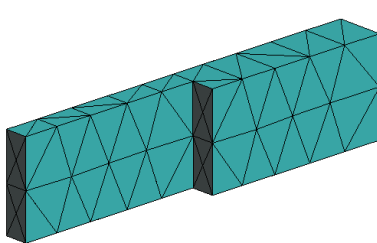
| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$ |
|---------------|----------------|------------|------------|------------|----------------|
| $0.3\lambda$  | 534            | 1.88       | 0.1582588  | 0.987397   | $17.05^\circ$  |
| $0.2\lambda$  | 855            | 2.75       | 0.1521252  | 0.988361   | $18.53^\circ$  |
| $0.1\lambda$  | 5732           | 8.90       | 0.1484998  | 0.988912   | $18.08^\circ$  |
| $0.09\lambda$ | 8516           | 15.64      | 0.1484410  | 0.988921   | $18.03^\circ$  |
| $0.08\lambda$ | 12616          | 24.26      | 0.1482657  | 0.988947   | $17.99^\circ$  |
| $0.07\lambda$ | 19384          | 46.91      | 0.1481798  | 0.988960   | $17.95^\circ$  |
| $0.06\lambda$ | 32063          | 124.87     | 0.1480343  | 0.988982   | $17.91^\circ$  |
| $0.05\lambda$ | 58145          | 392.87     | 0.1479451  | 0.988995   | $17.86^\circ$  |
| (MM)          | —              | —          | 0.1472465  | 0.989099   | $16.89^\circ$  |



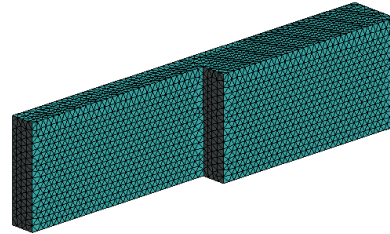
**Figura 3.15:** Convergencia del error para el cambio de ancho de la guía

Los resultados de las simulaciones se vuelven a tomar para un valor de frecuencia de  $K = 1.5 K_c$ , correspondiendo este valor a una frecuencia de 5.625 GHz. Al igual que en pruebas anteriores, la importancia del mallado en los resultados se comprueba generando diversos mallados con tamaños que van desde  $0.3 \lambda$  a  $0.05 \lambda$ . Las Fig. 3.16(a) y 3.16(b) muestran los mallados para los tamaños mencionados anteriormente.

La comparación de los resultados obtenidos por el código y por la técnica de MM se muestran en la Tabla 3.7. La Fig. 3.17 muestra la evolución del error y el tiempo de cómputo frente al mallado. A partir de  $0.2 \lambda$  los resultados de la simulación se asemejan bastante a los dados por MM. A partir de  $0.1 \lambda$  el error se encuentra por debajo del 1 %. Desde aquí la reducción del error es mínima, mientras que el tiempo de cómputo aumenta considerablemente, por lo que para simular esta estructura, un mallado de  $0.1 \lambda$  es suficiente para obtener resultados precisos y con un tiempo de cómputo pequeño. Esta elección en el mallado se indica por medio de la línea roja discontinua de la figura.



(a) Mallado  $0.3\lambda$



(b) Mallado  $0.05\lambda$

**Figura 3.16:** Cambio del alto de la guía

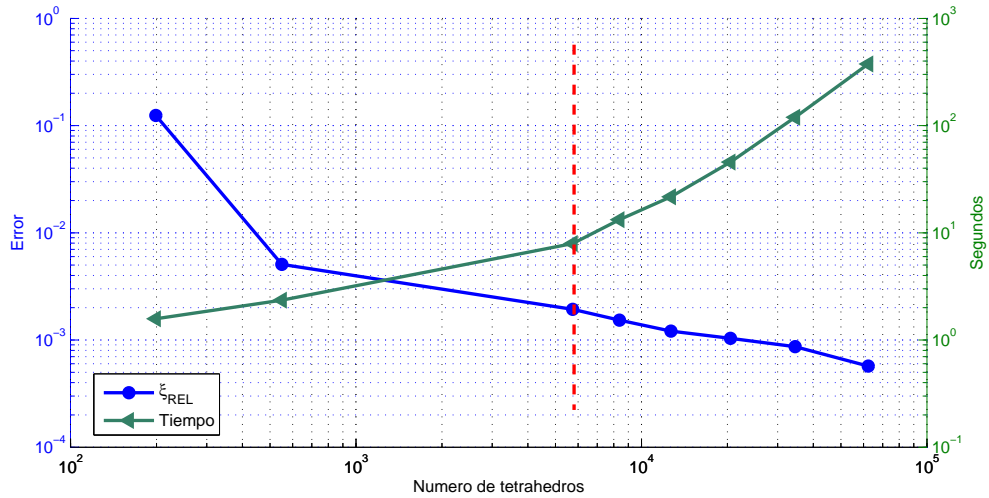
### 3.2. PROBLEMAS DE GUÍAS DE ONDAS

**Tabla 3.7:** Parámetros S del cambio del alto de la guía

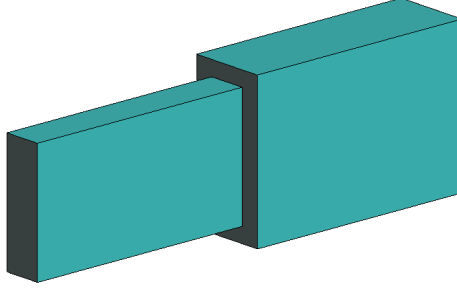
| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$ |
|---------------|----------------|------------|------------|------------|----------------|
| $0.3\lambda$  | 199            | 1.58       | 0.4317509  | 0.8135911  | $81.53^\circ$  |
| $0.2\lambda$  | 550            | 2.35       | 0.3812676  | 0.9244647  | $83.55^\circ$  |
| $0.1\lambda$  | 5740           | 7.94       | 0.3741039  | 0.9273867  | $84.64^\circ$  |
| $0.09\lambda$ | 8373           | 13.25      | 0.3731799  | 0.9277590  | $84.76^\circ$  |
| $0.08\lambda$ | 12684          | 21.69      | 0.3724377  | 0.9280571  | $84.86^\circ$  |
| $0.07\lambda$ | 20505          | 45.67      | 0.3720364  | 0.9282181  | $84.92^\circ$  |
| $0.06\lambda$ | 34537          | 119.61     | 0.3716455  | 0.9283747  | $84.97^\circ$  |
| $0.05\lambda$ | 62311          | 375.89     | 0.3709557  | 0.9286505  | $85.07^\circ$  |
| (MM)          | —              | —          | 0.3696254  | 0.9291808  | $85.10^\circ$  |

#### 3.2.7 Prueba 7: Cambio en el alto y ancho

Esta estructura es una guía rectangular que tiene un cambio simultáneamente en la altura y en el ancho (véase Fig. 3.18). Un tramo de guía está centrado respecto al otro, y se utiliza principalmente para conectar distintas secciones de guías. Se puede ver como una combinación de los dos ejemplos anteriores. Las dimensiones son tales que  $a_2 = 0.8 a_1$ ,  $\frac{a_1}{b_1} = 2$ ,  $\frac{b_2}{b_1} = 2$  y además las longitudes son  $l_1 = l_2 = 1.5 a_1$ . En este caso en particular, las medidas seleccionadas son  $a_1 = 4$  cm,  $a_2 = 3.2$  cm,  $b_1 = 2$  cm,  $b_2 = 1$  cm y  $l_1 = l_2 = 6$  cm. Como en todos los ejemplos anteriores los resultados de las simulaciones se han tomado para un valor de frecuencia de 5.625 GHz y diversos mallados han sido considerados. Los resultados de la simulación se muestran en la Tabla 3.8.



**Figura 3.17:** Convergencia del error para el cambio de alto de la guía



**Figura 3.18:** Cambio en el alto y en el ancho de la guía

La Fig. 3.19 muestra como disminuye el error a medida que aumenta el número de incógnitas del problema. A partir de  $0.2\lambda$  el error está en torno al 1% y continúa decreciendo. Por otro lado, es partir de  $0.9\lambda$  cuando el tiempo de cómputo crece de manera sustancial. Por lo que para simular esta estructura, un mallado de  $0.1\lambda$  es suficiente para obtener resultados con un error inferior al 1% y con un tiempo de cómputo pequeño.

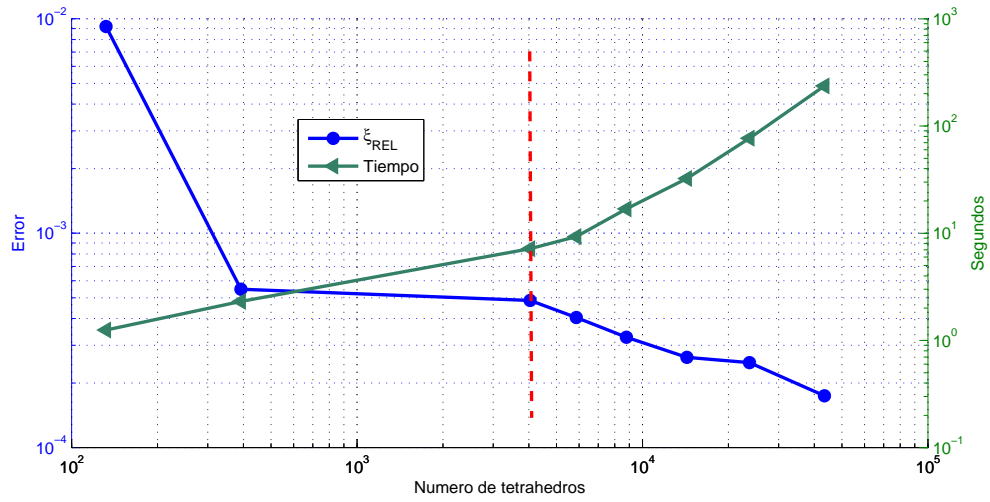
### 3.2.8 Prueba 8: T-Mágica

Esta estructura conocida como T-Mágica, es una combinación de dos uniones en T, una en el plano H y otra en el plano E. La T-Mágica es un tipo de acoplador direccional, utilizado principalmente como divisor o combinador de potencia. Cuando los puertos 3 y 4 están simultáneamente acoplados, la simetría y la unitariedad de la matriz de parámetros S implica que los puertos 3 y 4 están desacoplados. El aislamiento entre los puertos 3 y 4 lo podemos ver en la Fig. 3.20 donde se muestra la magnitud del campo magnético cuando es excitada por el puerto 3.

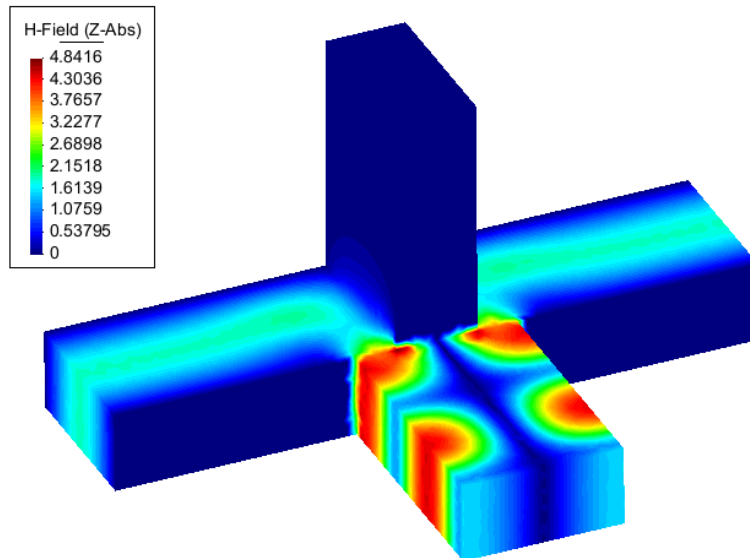
**Tabla 3.8:** Parámetros S del cambio en el alto y en el ancho de la guía

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{21})$ |
|---------------|----------------|------------|------------|------------|----------------|
| $0.3\lambda$  | 132            | 1.25       | 0.1650645  | 0.9862827  | $12.11^\circ$  |
| $0.2\lambda$  | 392            | 2.31       | 0.2143347  | 0.9767479  | $12.58^\circ$  |
| $0.1\lambda$  | 4043           | 7.18       | 0.2141116  | 0.9768092  | $12.56^\circ$  |
| $0.09\lambda$ | 5863           | 9.29       | 0.2137493  | 0.9768885  | $12.65^\circ$  |
| $0.08\lambda$ | 8792           | 16.86      | 0.2134029  | 0.9769642  | $12.78^\circ$  |
| $0.07\lambda$ | 14323          | 32.44      | 0.2131192  | 0.9770262  | $12.90^\circ$  |
| $0.06\lambda$ | 23718          | 77.00      | 0.2130537  | 0.9770404  | $12.79^\circ$  |
| $0.05\lambda$ | 43407          | 236.95     | 0.2127198  | 0.9771132  | $12.86^\circ$  |
| (MM)          | —              | —          | 0.2119345  | 0.9772838  | $13.00^\circ$  |

### 3.2. PROBLEMAS DE GUÍAS DE ONDAS



**Figura 3.19:** Convergencia del error para el cambio de alto y en el ancho de la guía



**Figura 3.20:** Módulo de la componente z del campo-H excitando el puerto 3.

### 3.2. PROBLEMAS DE GUÍAS DE ONDAS

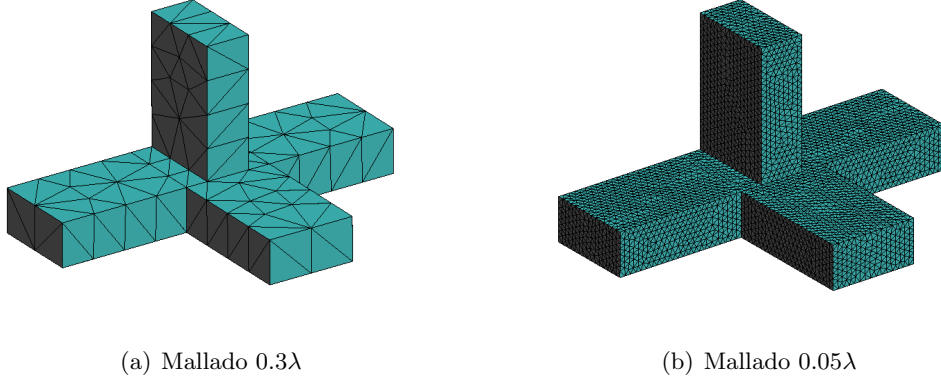
**Tabla 3.9:** Parámetros S de la T Mágica

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ |
|---------------|----------------|------------|------------|------------|
| $0.3\lambda$  | 295            | 0.53       | 0.0709646  | 0.5285525  |
| $0.2\lambda$  | 1279           | 1.33       | 0.0540885  | 0.5029231  |
| $0.1\lambda$  | 15242          | 15.408     | 0.0526138  | 0.5068732  |
| $0.09\lambda$ | 21082          | 87.11      | 0.0530391  | 0.5068105  |
| $0.08\lambda$ | 31731          | 185.75     | 0.0528693  | 0.5070763  |
| $0.07\lambda$ | 49677          | 313.34     | 0.0525706  | 0.5074737  |
| $0.06\lambda$ | 82690          | 712.795    | 0.0523698  | 0.5075709  |
| $0.05\lambda$ | 150672         | 5371.95    | 0.0525301  | 0.5078142  |
| (MM)          | —              | —          | 0.0521547  | 0.5083179  |

| $ S_{31} $ | $ S_{41} $ | $ S_{33} $ | $ S_{43} $ | $ S_{44} $ |
|------------|------------|------------|------------|------------|
| 0.5832603  | 0.6033581  | 0.5622714  | 0.0051892  | 0.5035999  |
| 0.5871605  | 0.6314628  | 0.5580313  | 0.0020778  | 0.4482081  |
| 0.6298554  | 0.5861483  | 0.4541678  | 0.0003122  | 0.5595873  |
| 0.5857029  | 0.6302867  | 0.5596822  | 0.0006004  | 0.4539484  |
| 0.5857893  | 0.6300127  | 0.5601879  | 0.0003452  | 0.4539778  |
| 0.5858089  | 0.6297048  | 0.5598912  | 0.0001661  | 0.45508126 |
| 0.5858757  | 0.6295828  | 0.5597984  | 0.0000911  | 0.4553784  |
| 0.5857749  | 0.6294684  | 0.5601806  | 0.0001326  | 0.4554953  |
| 0.5859     | 0.6290568  | 0.5599456  | $10^{-15}$ | 0.4568745  |

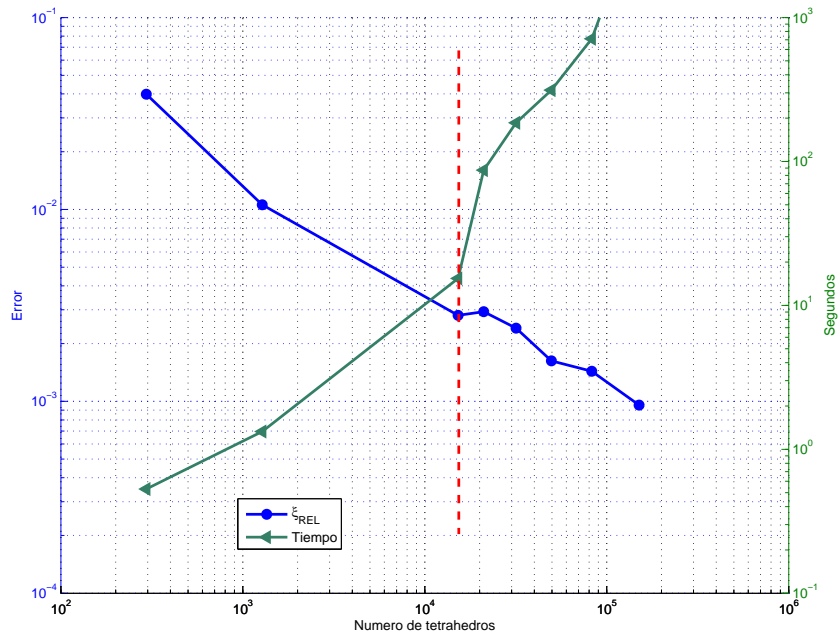
Además, en la figura se observa la componente longitudinal del campo en el puerto 3, y la componente transversal en los puertos 1 y 2. En el puerto 4 no hay campo. Las dimensiones de la estructura son tales que  $\frac{a}{b} = 2$ , y además las longitudes de las guías son  $l_1 = l_2 = l_3 = l_4 = 1.5a$ . Particularizando los valores de los parámetros  $a$  y  $b$  para este caso, se tiene que,  $a = 4$  cm,  $b = 2$  cm y  $l_1 = l_2 = l_3 = l_4 = 6$  cm.

Al igual que en todas las pruebas anteriores la frecuencia a la cual se obtienen los resultados es de 5.625 GHz. También se han generado varios mallados con diversos tamaños para ver su influencia en la precisión de la simulación. Véanse las Fig. 3.21(a) y 3.21(b) que muestran los mallados para los diversos tamaños. Los resultados de la simulación se muestran en la Tabla 3.9.



**Figura 3.21:** T Mágica

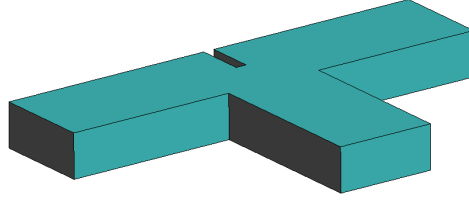
La Fig. 3.22 representa la evolución del error y del tiempo de cómputo en función del tamaño del mallado. Se puede ver como el error va disminuyendo conforme aumenta la discretización del mallado y es, a partir de  $0.2\lambda$ , cuando ya se obtienen unos valores muy aproximados en la mayoría de los parámetros, con un error en torno al 1 %. La diferencia de los resultados obtenidos con un tamaño de  $0.09\lambda$  y  $0.1\lambda$ , es muy pequeña, en cambio el tiempo de cómputo crece de forma considerable, por lo que para simular esta geometría, un mallado de  $0.1\lambda$  proporciona una relación muy buena de precisión y tiempo de cómputo, reflejado en la figura mediante una línea discontinua de color rojo.



**Figura 3.22:** Convergencia del error para la T-Mágica

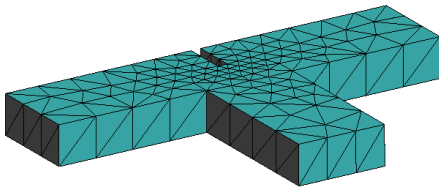
### 3.2.9 Prueba 9: Unión en T

La siguiente estructura bajo análisis es una unión en T con un tabique en el centro del lado mayor (véase la figura 3.23). Esta estructura se utiliza para dividir una señal entrante en dos señales de salida. El tabique divide la señal y la dirige a los puertos de salida, al tiempo que minimiza la reflexión en el puerto de entrada. Las medidas de la estructura son  $a = 22.866$  mm,  $b = 10.163$  mm y  $L = 39.38$  mm para cada uno de los brazos de la unión. El tabique está situado centrado en frente del puerto 1 a una distancia  $l_2 = 49.5425$  mm de los puertos 2 y 3. Tiene unas dimensiones de  $t = 11.433$  mm de ancho y  $h = 2.5406$  mm de espesor. Para las medidas indicadas anteriormente, se tiene una frecuencia de trabajo de 10 GHz.

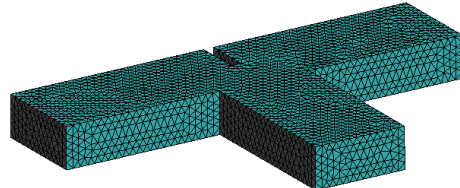


**Figura 3.23:** Unión en T

Para ver la importancia del mallado en los resultados se han generado diversos mallados con tamaños que van desde  $0.3 \lambda$  a  $0.06 \lambda$ . Las figuras 3.24(a) y 3.24(b) muestran los mallados para los tamaños mencionados anteriormente. Los resultados de la simulación de esta estructura se muestran en la tabla 3.10. La figura 3.25 muestra la evolución del error y del tiempo de cómputo conforme aumentan el número de incógnitas. En las simulaciones de  $0.3 \lambda$  a  $0.1 \lambda$  se observa como los valores se van aproximando de manera progresiva a los obtenidos mediante el software comercial HFSS. Los resultados de la simulación para un tamaño de  $0.09 \lambda$  prácticamente permanecen invariantes con respecto a los resultados de la simulación a  $0.1 \lambda$ , por lo que para simular esta estructura, el mallado de  $0.1 \lambda$  proporciona una relación muy buena de precisión y tiempo de cómputo, marcado en la figura por una línea de color rojo. Adicionalmente puede observarse como a partir de  $0.08 \lambda$ , la pendiente del tiempo de cómputo aumenta considerablemente, y la del error decae levemente.



(a) Mallado 0.3



(b) Mallado 0.06

**Figura 3.24:** Mallados de la unión en T.

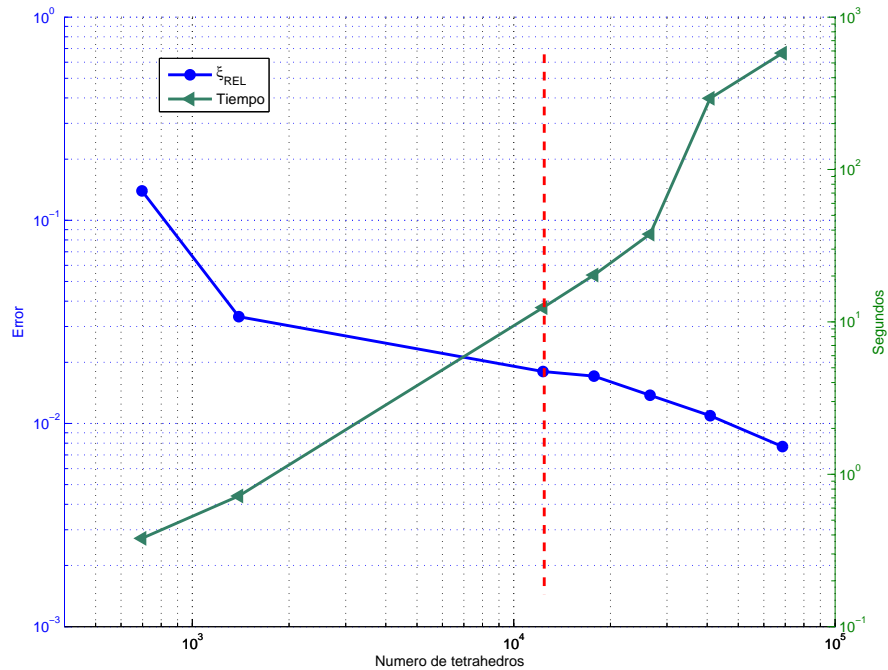


### 3.2. PROBLEMAS DE GUÍAS DE ONDAS

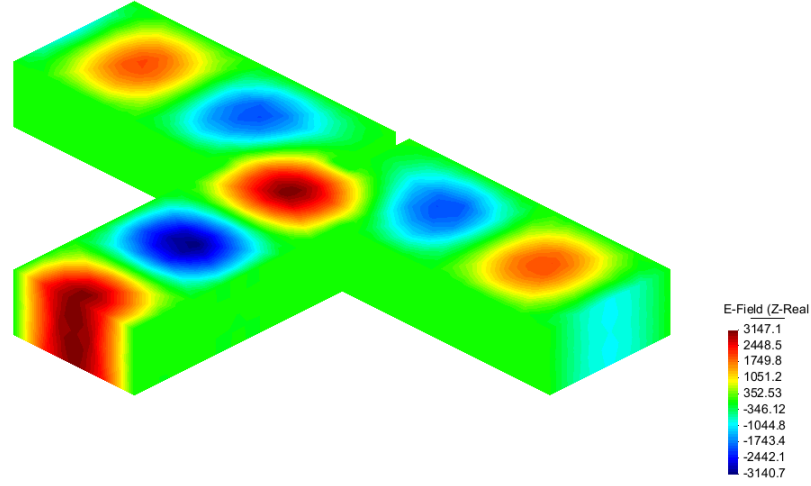
**Tabla 3.10:** Parámetros S de la unión en T con tabique centrado

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{11} $ | $ S_{21} $ | $\arg(S_{31})$   |
|---------------|----------------|------------|------------|------------|------------------|
| $0.3\lambda$  | 697            | 0.38       | 0.166386   | 0.692636   | $-118.512^\circ$ |
| $0.2\lambda$  | 1395           | 0.72       | 0.199868   | 0.692373   | $-116.83^\circ$  |
| $0.1\lambda$  | 12339          | 12.411     | 0.196868   | 0.693209   | $-117.40^\circ$  |
| $0.09\lambda$ | 1778           | 20.28      | 0.196692   | 0.693342   | $-117.45^\circ$  |
| $0.08\lambda$ | 26572          | 37.651     | 0.196047   | 0.693342   | $-117.53^\circ$  |
| $0.07\lambda$ | 40890          | 293.18     | 0.195497   | 0.693466   | $-117.60^\circ$  |
| $0.06\lambda$ | 68587          | 580.62     | 0.194872   | 0.693544   | $-117.68^\circ$  |
| (HFSS)        | —              | —          | 0.193385   | 0.693753   | $-117.856^\circ$ |

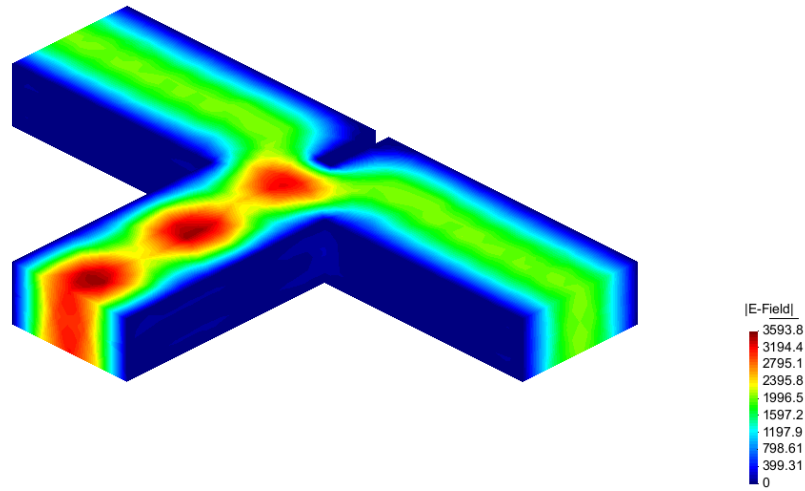
La figura 3.26 muestra la parte real de la componente z del campo eléctrico. En esta figura se aprecia claramente la sucesión de máximos y mínimos a lo largo de la sección de guía del puerto 1, al llegar al tabique la señal se divide entre los dos brazos de salida, dando lugar a una nueva sucesión de máximos y mínimos pero con la mitad de intensidad de campo. La figura 3.27 representa el módulo del campo eléctrico. En dicha figura se muestra como el campo en los brazos de salida es la mitad con respecto al campo en la entrada.



**Figura 3.25:** Convergencia del error para la unión en T



**Figura 3.26:** Parte real de componente z del campo eléctrico.



**Figura 3.27:** Magnitud de campo  $|E_{total}|$ .

### 3.2.10 Prueba 10: Combinador de guías de ondas

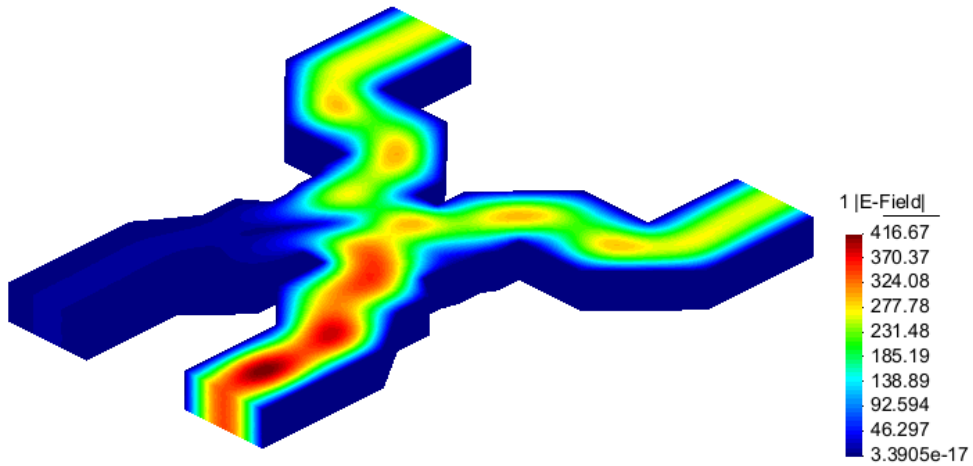
La última estructura de esta batería de pruebas es un combinador de guías de ondas de 4 puertos comercial modelo WR42 estándar. Cada guía mide 420 milésimas de pulgadas de ancho y 170 de alto. Este tipo de combinador de guías de onda se usa para combinar la potencia de salida de dos amplificadores de potencia de estado sólido de 20 GHz (SSPA) con pequeño tamaño y bajas pérdidas de inserción. La figura 3.28 muestra el combinador simulado. Siguiendo el mismo esquema que en las pruebas realizadas anteriormente, los resultados de las simulaciones se han tomado para un solo valor de frecuencia (20 GHz en este caso), además de generar diversos mallados con tamaños que van desde  $0.3$  a  $0.06 \lambda$ .

### 3.2. PROBLEMAS DE GUÍAS DE ONDAS

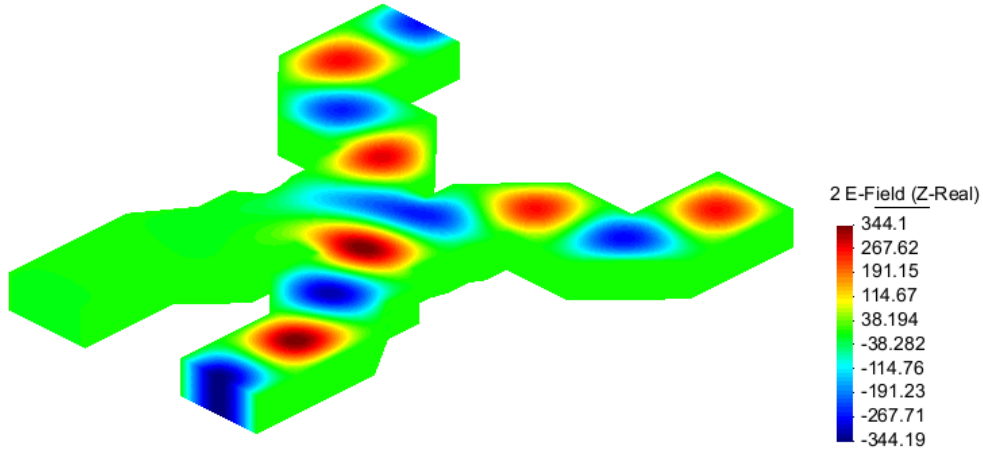
**Tabla 3.11:** Parámetros S del combinador

| Tamaño        | Num. elementos | Tiempo (s) | $ S_{21} $ | $ S_{31} $ | $\arg S_{21} $ |
|---------------|----------------|------------|------------|------------|----------------|
| $0.3\lambda$  | 1023           | 0.576      | 0.698376   | 0.705199   | $91.44^\circ$  |
| $0.2\lambda$  | 2282           | 1.18       | 0.703559   | 0.703072   | $88.60^\circ$  |
| $0.1\lambda$  | 22059          | 25.61      | 0.700952   | 0.705954   | $86.72^\circ$  |
| $0.09\lambda$ | 31348          | 179.446    | 0.700492   | 0.706509   | $86.56^\circ$  |
| $0.08\lambda$ | 45584          | 209.06     | 0.700266   | 0.706744   | $86.44^\circ$  |
| $0.07\lambda$ | 71482          | 423.153    | 0.699971   | 0.707084   | $86.35^\circ$  |
| $0.06\lambda$ | 120180         | 1202.616   | 0.699843   | 0.707224   | $86.28^\circ$  |
| (HFSS)        | —              | —          | 0.699295   | 0.707864   | $86.12^\circ$  |

La tabla 3.11 muestra los valores de los parámetros S del combinador cuando es excitado por el puerto 1 con una señal a 20 GHz. Los resultados de la simulación son muy cercanos a los resultados obtenidos con el software HFSS con lo que el correcto funcionamiento del código queda demostrado con esta simulación de un combinador comercial. La figura 3.28 representa el módulo del campo eléctrico cuando el puerto 1 está excitado. El campo eléctrico en los brazos de salida 2 y 4 (los dos situados en la parte superior) es la mitad que el campo en el brazo de entrada (el puerto situado en la parte inferior derecha) y que el brazo 3 permanece aislado ya que el valor del campo es prácticamente nulo. La figura 3.29 muestra la parte real de la componente z del campo eléctrico cuando el puerto 1 está excitado. En dicha figura se puede observar como el campo en los brazos de salida 2 y 4 está desfasado 90 grados.



**Figura 3.28:** Magnitud de campo  $|E_{total}|$ .



**Figura 3.29:** Parte real de la componente z del campo eléctrico del combinador a 20 GHz.

#### 3.2.11 Conclusiones

Tras realizar el análisis de estas 10 estructuras podemos concluir que los resultados obtenidos en las simulaciones vienen condicionados por el mallado generado. Los resultados van mejorando a medida que aumenta el mallado, llegando a ser muy precisos para tamaños de mallado por debajo de  $0.08 \lambda$ .

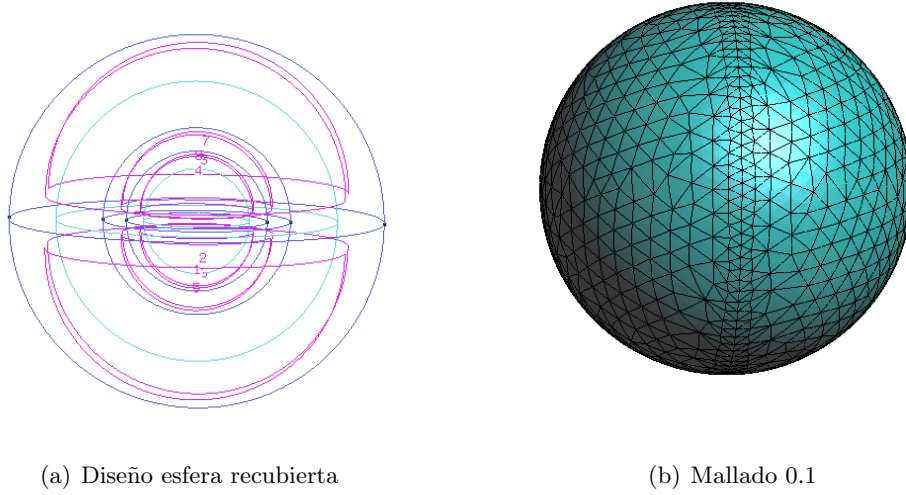
En contraposición a la precisión de los resultados, a medida que el mallado es más fino también aumenta el tiempo de cómputo. A partir de  $0.1 \lambda$ , el aumento en el tiempo de cómputo es más significativo que la mejoría obtenida en los resultados, por lo que para realizar el análisis de estas estructuras no es necesario aumentar más el mallado salvo que el usuario quiera obtener una mayor precisión.

La principal conclusión que se puede obtener de esta primera batería de pruebas es el que código ha pasado todas la verificaciones básicas y que, además, el tamaño del mallado necesario para obtener unas simulaciones con un tiempo de cómputo adecuado y una precisión aceptable debe ser de aproximadamente  $0.1 \lambda$ .

### 3.3 Problemas de radiación/scattering

Esta serie de pruebas va a consistir en el análisis de problemas llamados coloquialmente abiertos, es decir, de problemas de radiación y de problemas de scattering. Con esta batería de pruebas se va a probar la parte del código que realiza el truncamiento de malla necesario para resolver este tipo de problemas, así como, la parte del código que los excita.

El código posee dos métodos de truncamiento de malla que van a ser testados mediante el análisis de tres estructuras diferentes: una esfera metálica recubierta por un dieléctrico, un objeto metálico y una antena. El primer método de truncamiento de malla consiste



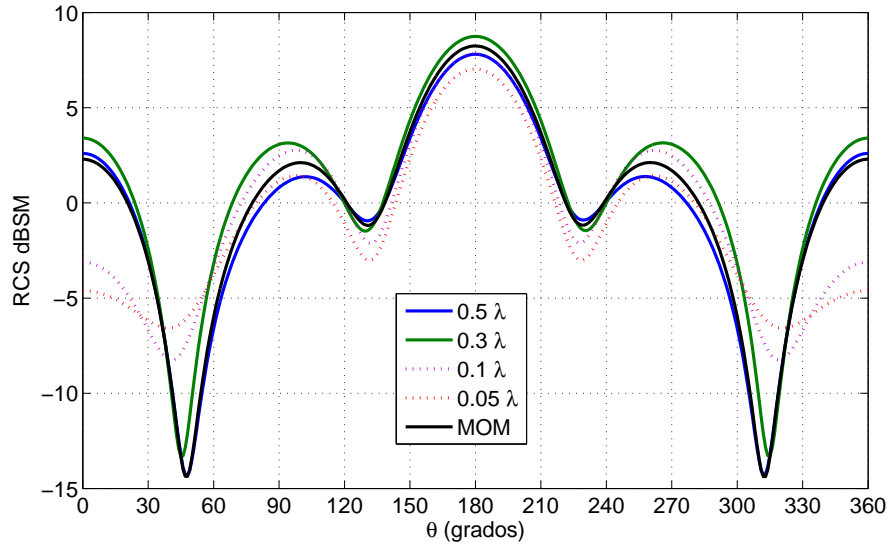
**Figura 3.30:** Mallado de la esfera recubierta  $0.1\lambda$ .

simplemente en la asignación de una condición de contorno absorbente de primer orden (ABC de primer orden) a la frontera que trunca la malla. Esta frontera se situará a diferentes distancias comprobando el comportamiento del código ante dichos cambios. El segundo método de truncamiento de malla consiste en la aplicación de un método integral llamado *evaluación iterativa de ecuación integral* o método IIEE a la frontera que trunca la malla. Este segundo método permite acercar la frontera externa al objeto de manera que se reduce el tamaño del problema. De esta forma, lo que se pretende con esta batería de pruebas es ver hasta donde se puede acercar dicha frontera manteniendo la precisión y utilizando un tiempo de cómputo reducido.

### 3.3.1 Prueba 11: Análisis biestático de esfera recubierta

Esta estructura consiste en una esfera metálica recubierta por un material dieléctrico. El radio de la esfera interior es de  $r_1 = 0.3$  m, el radio de la esfera exterior es de  $r_2 = 0.4$  m y la permitividad relativa del material del recubrimiento respecto al vacío es  $E_r = 3$ . La Fig. 3.30 muestra la geometría y el mallado de la estructura propuesta para esta prueba. La esfera es iluminada con una onda plana incidente sobre el eje  $z$  (polarización  $\theta\theta$ ) a la frecuencia de 300 MHz.

En primer lugar se va a utilizar la condición ABC de primer orden para realizar el truncamiento de malla. La distancia de la frontera al objeto varía desde  $0.05\lambda$  hasta  $0.5\lambda$  para ver el comportamiento del código conforme varía esta distancia. En lo relativo al tamaño del mallado, en esta sección se va a utilizar el obtenido como conclusión de la primera batería de pruebas, es decir se utilizará en todas las simulaciones  $0.1\lambda$ . La Fig. 3.31 muestra el corte en azimuth de la sección radar (RCS) al ir variando la distancia de la frontera al objeto, manteniendo un mallado de  $0.1\lambda$ .



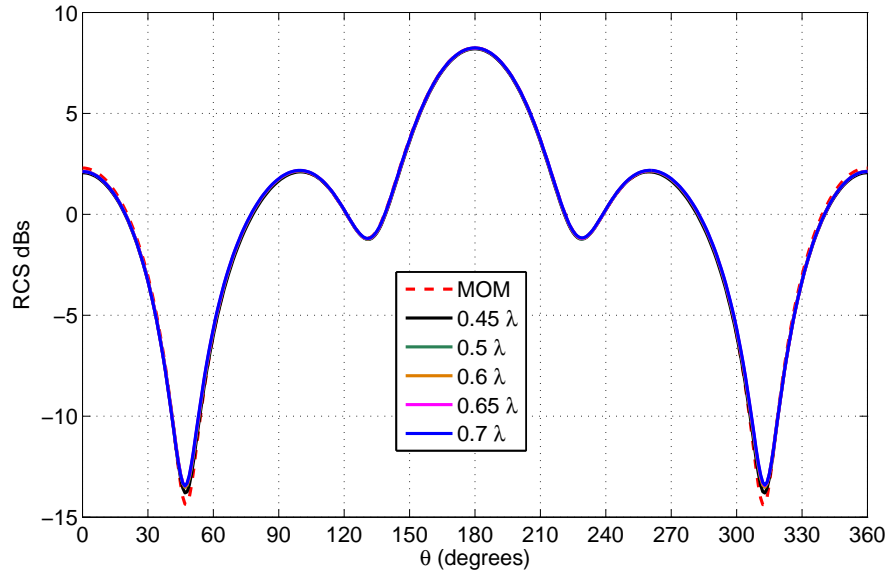
**Figura 3.31:** RCS bistática de la esfera para un mallado de  $0.1 \lambda$

A la vista de estos resultados se puede observar como a medida que la distancia de la frontera al objeto aumenta, los resultados se aproximan más a los obtenidos mediante el método de los momentos (MOM) (considerado de referencia para este caso), tal y como era esperado.

La Tabla 3.12 muestra la evolución del tiempo de cómputo a medida que alejamos la frontera del objeto, de tal forma que a mayor distancia, el número de elementos generados al mallar es mayor, aumentado de esta forma el tamaño del problema, de manera que el tiempo de cómputo también aumenta. De esta forma, para obtener un resultado preciso usando la condición ABC de primer orden, la frontera al objeto ha de situarse a una distancia mayor de  $0.4\lambda$  incrementando el tamaño del problema. La pregunta que nos podemos hacer ahora es, si el segundo método de truncamiento de malla (método IIEE) nos permite reducir el tamaño del problema hasta tal punto que el tiempo de cómputo necesario para resolver el problema sea menor que usando la condición ABC de primer orden.

**Tabla 3.12:** Comparativa de tiempo frente a mallado

| Tamaño         | Elementos | Tiempo (s) |
|----------------|-----------|------------|
| $0.5 \lambda$  | 35567     | 41.013     |
| $0.4 \lambda$  | 25188     | 22.169     |
| $0.3 \lambda$  | 17880     | 13.782     |
| $0.2 \lambda$  | 11828     | 6.162      |
| $0.1 \lambda$  | 7237      | 2.995      |
| $0.05 \lambda$ | 5465      | 2.215      |



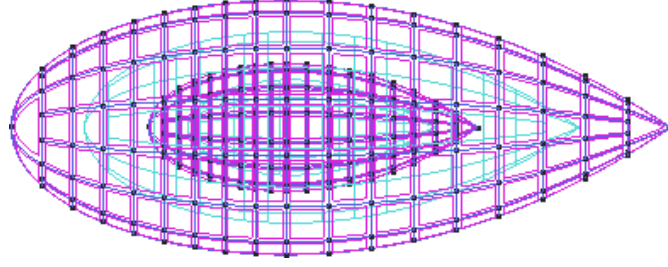
**Figura 3.32:** RCS biestática de la esfera para un mallado de  $0.1 \lambda$

A continuación se va a utilizar el segundo método de truncamiento de malla para resolver el mismo problema variando la distancia de la frontera al objeto desde  $0.3\lambda$  hasta  $0.05\lambda$ . En lo relativo al tamaño del mallado se va a utilizar el mismo que anteriormente, es decir,  $0.1\lambda$ . La Fig. 3.32 muestra el corte en azimut de la RCS para las diferentes distancias mencionadas anteriormente. Se puede ver como utilizando el método IIEE, los resultados son prácticamente idénticos independientemente de la distancia de la frontera al objeto. Sin embargo, el tiempo y el número de iteraciones requeridas para obtener estos resultados sí que varían en función de la proximidad de la frontera externa.

La Tabla 3.13 muestra los datos de tiempo e iteraciones dependiendo de la distancia de la frontera al objeto. A la vista de esos resultados, podemos ver como a medida que nos aproximamos al objeto, el tamaño del problema se reduce, reduciéndose también el tiempo de cómputo, ya que aunque aumenta el número de iteraciones necesarias para resolver el problema, el tiempo necesario en cada una de ellas es menor.

**Tabla 3.13:** Comparativa Memoria tiempo de cómputo

| Distancia     | Elementos | Incógnitas | Iteraciones | Tiempo total (s) |
|---------------|-----------|------------|-------------|------------------|
| $0.05\lambda$ | 5465      | 38260      | 10          | 26.161           |
| $0.1\lambda$  | 7237      | 49766      | 8           | 24.032           |
| $0.2\lambda$  | 11838     | 80022      | 7           | 33.14            |
| $0.25\lambda$ | 14300     | 96414      | 6           | 36.963           |
| $0.3\lambda$  | 17880     | 119546     | 5           | 38.232           |



**Figura 3.33:** NASA Almond

Esto se cumple hasta que llegamos a un punto que si continuamos aproximando la frontera al objeto, el número de iteraciones crece en mayor medida que lo que se reduce el tiempo de cada iteración, por lo que el tiempo total de ejecución aumenta. De esta forma, podemos concluir que la distancia óptima para emplazar la frontera estará en el entorno de  $0.1 \lambda$ .

### 3.3.2 Prueba 12: Análisis monoestático de la NASA Almond

Para continuar con las pruebas del código, lo siguiente a realizar será el análisis monoestático de un objeto metálico llamado NASA Almond por su forma de almendra. La geometría de esta estructura es bien conocida y está definida por fórmulas paramétricas que pueden ser encontradas, por ejemplo, en [21] y son mostradas a continuación.

|   |  |
|---|--|
| <p>for <math>-0.4167 &lt; t &lt; 0</math> and <math>-\pi &lt; \psi &lt; \pi</math></p> <p><math>x = d \cdot t</math> inches, <math>d = 9.936</math> inches</p> <p><math>y = 0.13333 \cdot d \cdot \sqrt{1 - \left(\frac{t}{0.416667}\right)^2} \cos \psi</math></p> <p><math>z = 0.064444 \cdot d \cdot \sqrt{1 - \left(\frac{t}{0.416667}\right)^2} \sin \psi</math></p> | <p>for <math>0 &lt; t &lt; 0.58333</math> and <math>-\pi &lt; \psi &lt; \pi</math></p> <p><math>x = d \cdot t</math> inches, <math>d = 9.936</math> inches</p> <p><math>y = 4.83345 \cdot d \cdot \left[ \sqrt{1 - \left(\frac{t}{2.08335}\right)^2} - 0.96 \right] \cos \psi</math></p> <p><math>z = 1.91115 \cdot d \cdot \left[ \sqrt{1 - \left(\frac{t}{2.08335}\right)^2} - 0.96 \right] \sin \psi</math></p> |
|---|--|

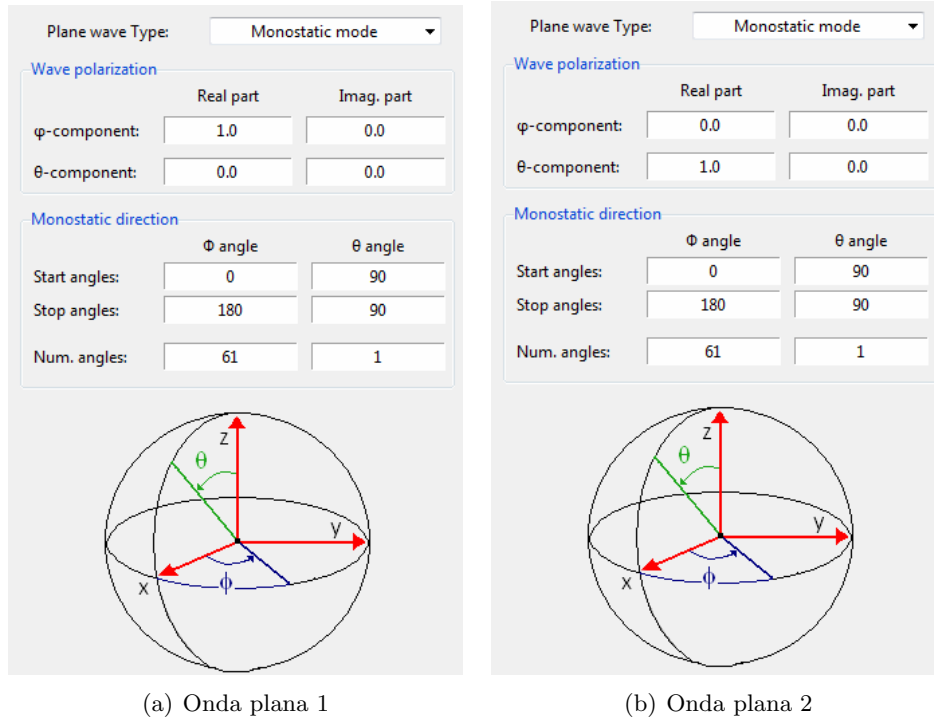
**Figura 3.34:** Descripción matemática de la NASA Almond

Al igual que en el caso anterior, el truncamiento del objeto se va a realizar de manera conforme, y aplicando, primeramente, una condición ABC de primer orden sobre la frontera con forma de almendra a una distancia de  $0.7 \lambda$  y de  $0.2 \lambda$ . Tomando como referencia el ejemplo anterior, podemos intuir que a una distancia de  $0.2 \lambda$  y usando la condición ABC de primer orden, los resultados van a diferir de los resultados que vamos a usar como referencia en la comparación. Dichos resultados son los obtenidos mediante una simulación realizada con MoM como en la prueba anterior.

La Fig. 3.33 muestra la NASA Almond creada según las fórmulas descritas anteriormente, junto con otra almendra concéntrica que defina la frontera externa. Tal y como indican las fórmulas, la almendra tiene una longitud de 9.936 pulgadas que a una frecuencia de trabajo de 1.19 GHz equivale aproximadamente a un tamaño eléctrico de  $\lambda$ .



### 3.3. PROBLEMAS DE RADIACIÓN/SCATTERING

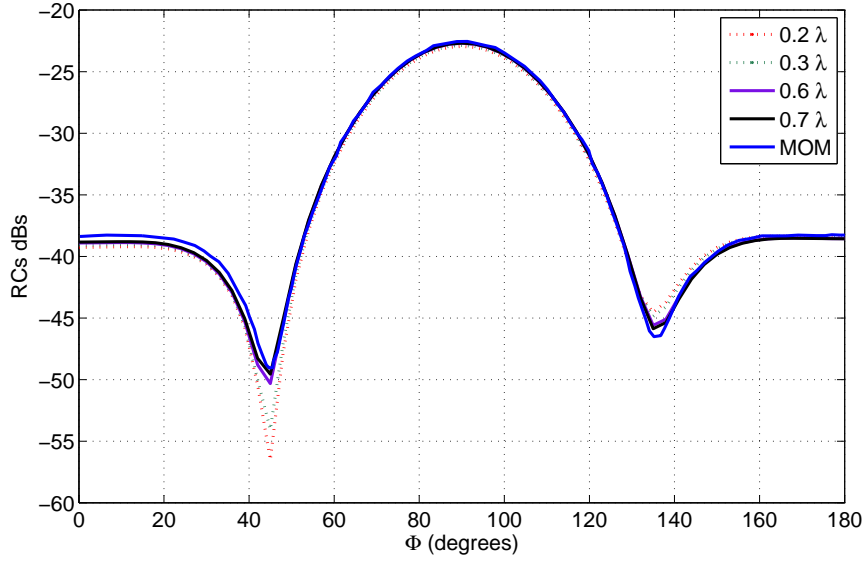


**Figura 3.35:** Detalles de la excitación monoestática

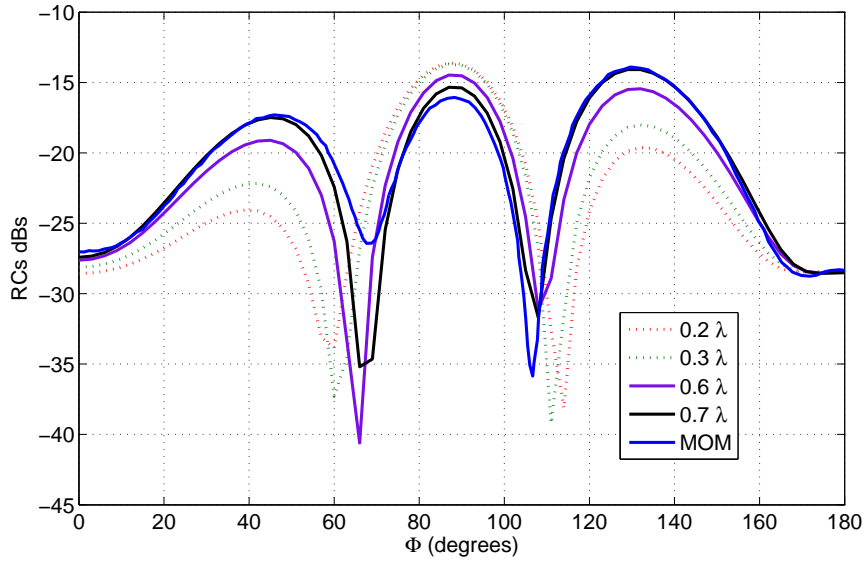
La excitación consistirá en ondas planas en el plano XY con polarización vertical y con polarización horizontal (véanse las Figs. 3.35(a) y 3.35(b) para más detalles sobre las propiedades de las ondas planas). Tanto para las ondas con polarización vertical como horizontal, se han escogido 61 en el rango entre  $\phi = 0$  y  $\phi = 180$  grados.

La Fig. 3.36 muestra la RCS para el caso monoestático con polarización vertical a los mencionados 1.19 GHz. En este caso, los resultados para las diferentes posiciones de la frontera presentan una gran similitud con los resultados de la simulación MoM. En cambio, observando la Fig. 3.37 vemos que los resultados de la RCS para el caso monoestático con polarización horizontal para valores próximos de la frontera distan bastante de los resultados de la simulación MoM. Conforme se va alejando la frontera, los resultados empiezan a mejorar, siendo a partir de una distancia de  $0.7\lambda$  cuando se empieza a tener un buen comportamiento. Esto hace que el tamaño del problema sea muy grande y por lo tanto los tiempos de ejecución aumenten, como puede verse en la Tabla 3.14.

Continuando con esta prueba, es hora de utilizar el método IIEE como truncamiento y comparar sus resultados con los obtenidos mediante la ABC de primer orden. Tal y como se puede observar en la prueba anterior, los resultados obtenidos con el método IIEE son siempre precisos independientemente de la distancia de la frontera. En cambio, el tiempo de la simulación y el número de iteraciones necesarias para que el método converja sí dependen fuertemente de la distancia utilizada. En esta prueba en particular, se van



**Figura 3.36:** RCS monoestática para polarización vertical

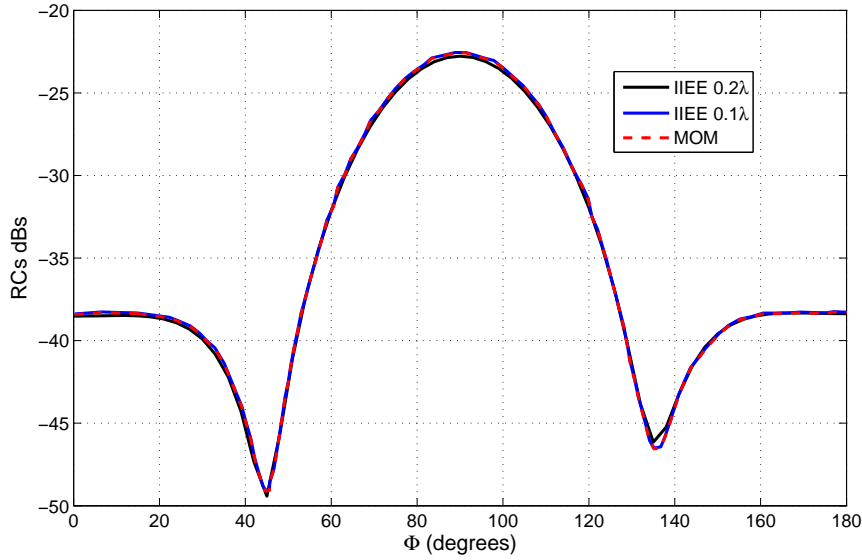


**Figura 3.37:** RCS monoestática para polarización horizontal

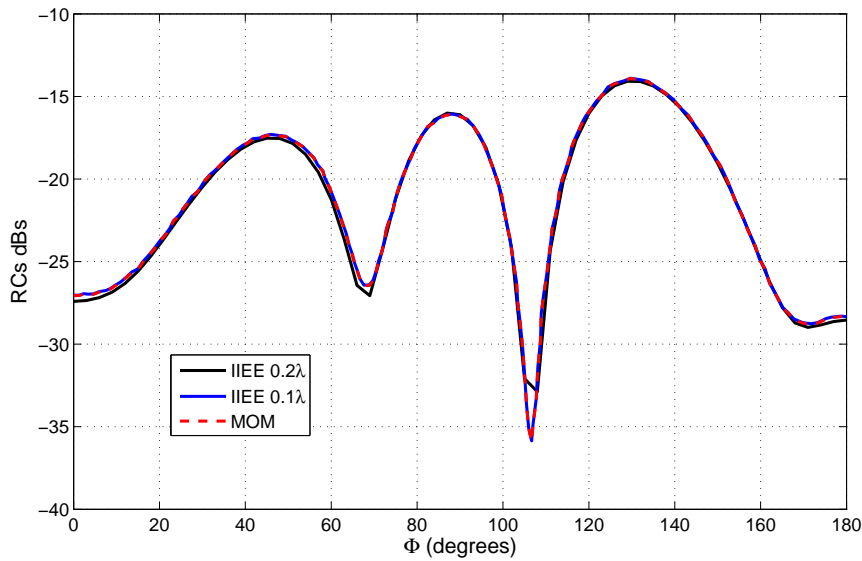
a utilizar unas distancias de  $0.2\lambda$  y  $0.1\lambda$ , respectivamente.

De este modo, se verificará el correcto funcionamiento del código para estos dos casos que no son precisos utilizando una ABC de primer orden y, a su vez, se comprobará si la reducción en el tamaño del problema al colocar la frontera mas cerca de la almendra, se corresponde con una reducción en el tiempo de simulación.

A la vista de las Figs. 3.38 y 3.39, vemos como los resultados dados por el método IIEE coinciden prácticamente con los obtenidos mediante la simulación MoM para las



**Figura 3.38:** RCS monoestática para polarización vertical usando IIEE



**Figura 3.39:** RCS monoestática para polarización horizontal usando IIEE

dos polarizaciones, con lo que se confirma el comportamiento esperado por el código en cuanto a la precisión del método.

Respecto al tiempo empleado en las simulaciones, los datos de todas ellas se muestran en la Tabla 3.14. Comprobamos como, para distancias cercanas, el tiempo de simulación para el método ABC es muy reducido, si bien hemos comprobado que los resultados no son buenos. Si tomamos como resultados aceptables los que obtenemos a una distancia de  $0.7\lambda$ , el tamaño del problema y el tiempo de ejecución se elevan en demasía llegando a tardar casi un 600 % más que en el caso de  $0.6\lambda$ . En el caso del truncamiento mediante

**Tabla 3.14:** Comparativa Memoria tiempo de cómputo

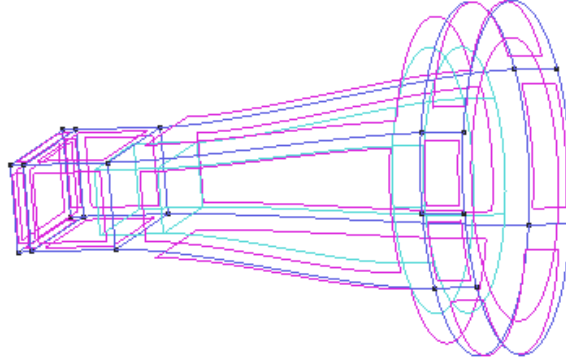
| Método | Distancia    | Elementos | Incógnitas | Tiempo (s) |
|--------|--------------|-----------|------------|------------|
| ABC    | $0.2\lambda$ | 26921     | 188772     | 36.239     |
| ABC    | $0.3\lambda$ | 38655     | 254204     | 54.215     |
| ABC    | $0.6\lambda$ | 46470     | 304982     | 792.185    |
| ABC    | $0.7\lambda$ | 56225     | 369280     | 4522.765   |
| IIEE   | $0.2\lambda$ | 43236     | 292116     | 30686      |
| IIEE   | $0.1\lambda$ | 28105     | 189886     | 42314      |

el método IIEE, los resultados presentan un gran precisión pero el tiempo empleado en la simulación, al tener muchas excitaciones, se dispara notablemente.

Como conclusión a esta prueba, vemos como la ABC de primer orden proporciona resultados precisos cuando la frontera está a distancias mayores de  $0.7\lambda$ , con el consiguiente aumento en el tamaño del problema y tiempo de simulación. En cambio, el método IIEE proporciona resultados muy precisos sin importar la distancia de la frontera. Sin embargo, debido al número de iteraciones y de excitaciones presentes en un problema monoestático, el tiempo de simulación usando este método de truncamiento es muy lento. Aquí entra la decisión del usuario entre una simulación más o menos precisa en un tiempo razonable, o una simulación muy lenta pero muy precisa y fiable como la que proporciona el método IIEE. Cabe destacar que la paralelización del código es excelente (en torno a un 75 % de rendimiento en paralelo) y a medida que se aumenta el número de CPU disponibles en el sistema, el tiempo de simulación se reduce. Por lo que, para concluir esta prueba, decir que si un usuario desea simular un análisis monoestático utilizando el método IIEE para obtener un resultados muy precisos, la mejor opción es utilizar un clúster HPC que le permita reducir el tiempo de simulación.

### 3.3.3 Prueba 13: Análisis de una bocina circular

Para finalizar con esta batería de pruebas, se va a realizar el análisis de una antena. En este caso se trata de una bocina con apertura circular. Las antenas de bocina son muy utilizadas en las bandas de frecuencia de microondas porque proporcionan alta ganancia, baja onda estacionaria, ancho de banda relativamente grande y son relativamente fáciles de construir. Además los cálculos teóricos concuerdan de manera muy exacta con las medidas de sus parámetros eléctricos. La apertura de la bocina será una circunferencia de radio  $r = 25,4$  mm. La excitación de la antena se realiza por medio de una guía rectangular de dimensiones  $a = 12,7$  mm,  $b = 25,4$  mm y estará situada a una distancia  $l = 63,5$  mm de la apertura de la bocina. La frecuencia de excitación será, en este caso, de 10 GHz. La Fig. 3.40 muestra la bocina bajo análisis conforme a las dimensiones indicadas.



**Figura 3.40:** Bocina circular

El procedimiento seguido será similar al realizado en el ejemplo anterior. En primer lugar se establece una frontera conforme al objeto y se utiliza como método de truncamiento de malla el ABC de primer orden, e iremos variando la distancia de la frontera al objeto.

La Fig. 3.41 muestra la directividad de la bocina, tomando el corte  $\Phi = 0$ . En este caso, los resultados muestran gran similitud para el haz principal (rango  $[-60, 60]$  grados) pero que difieren para el resto respecto a los resultados de la simulación MoM. En cambio, observando la Fig. 3.42 vemos que los resultados de la directividad para el corte  $\Phi = 90$  son bastante similares y cada vez más próximos conforme alejamos la frontera. Por contra, a medida que alejamos la frontera del objeto, el tamaño del problema aumenta y por tanto también lo hacen los tiempos de ejecución. Todos estos tiempos pueden verse en la Tabla 3.15.

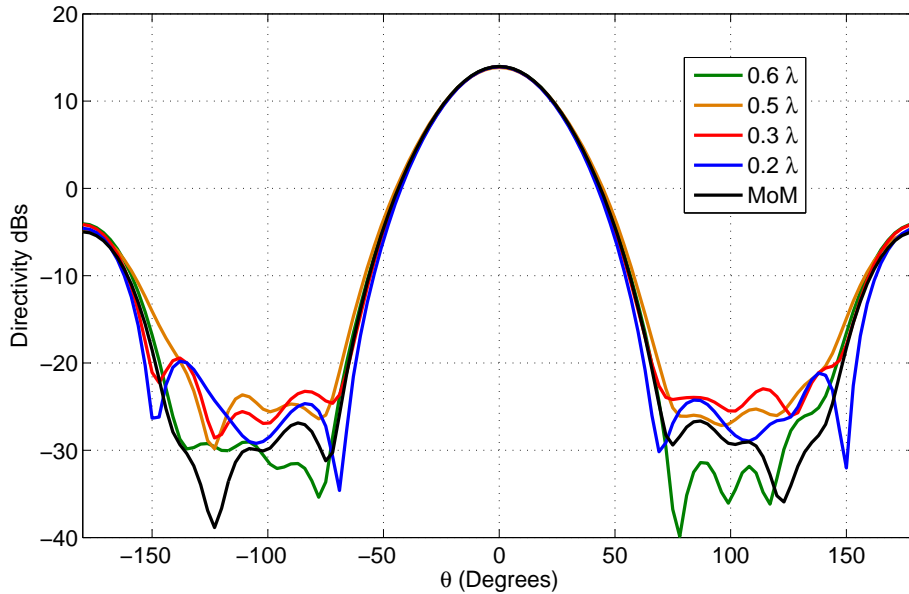
Continuamos la prueba utilizando el método IIEE como truncamiento. Al igual que hemos visto en las pruebas anteriores, los resultados utilizando este método, son mucho más precisos que los obtenidos mediante el ABC de primer orden. Estos resultados se muestran en la Fig. 3.43 y la Fig. 3.44, respectivamente. Las diferencias entre los resultados de las distintas simulaciones son casi idénticos, confirmando las conclusiones obtenidas ya desde las pruebas anteriores. Para problemas abiertos con pocas excitaciones, como pueden ser los de antenas, la mejor opción en cuanto a tiempo y precisión es aplicar el método IIEE a una distancia entre  $0.2$  y  $0.1\lambda$ .

#### 3.3.4 Conclusiones

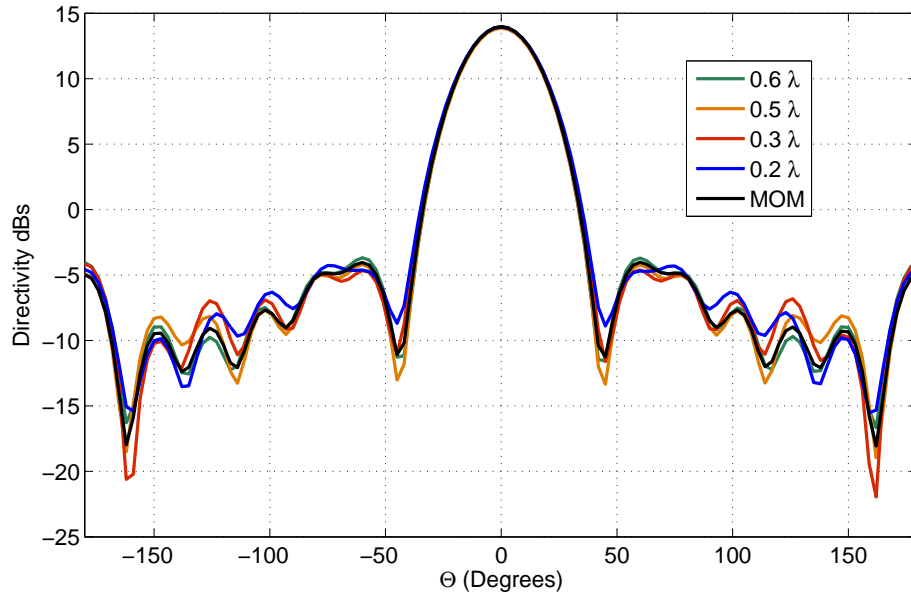
Tras realizar el análisis de estas 3 estructuras, vamos a resumir las conclusiones obtenidas en dichas pruebas. El uso de la condición ABC de primer orden para realizar el truncamiento de malla viene condicionado por la distancia a la que la frontera externa esté situada con respecto al objeto bajo estudio. Para conseguir resultados precisos es necesario colocar esta frontera a distancias mayores de  $0.45\lambda$ , con el consiguiente

aumento en el tamaño del problema y el tiempo de computación.

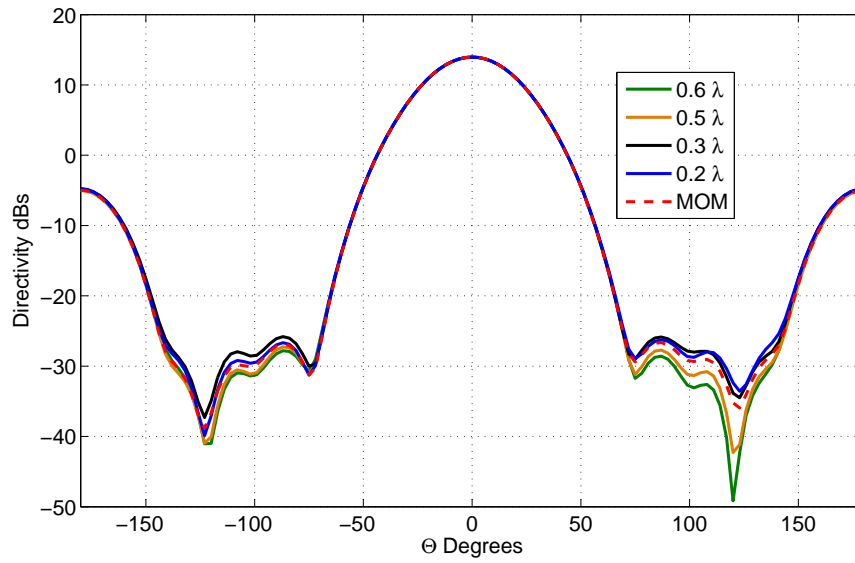
Por otro lado, la precisión de los resultados obtenidos mediante el uso del método IIEE para realizar el truncamiento de malla son independientes de la distancia a la que la frontera externa este situada. Sin embargo, el tiempo y el número de iteraciones requeridas para obtener estos resultados sí que varían en función de la proximidad de la frontera externa. En problemas en los que el número de excitaciones es pequeño, como pueden ser los problemas de antenas o análisis RCS biestáticos de objetos, el uso del método IIEE colocando la frontera externa a una distancia entre  $0.2\lambda$  y  $0.1\lambda$ , es la opción más precisa y rápida. En cambio, debido al gran número de excitaciones presentes a la hora de realizar un análisis RCS monoestático, el tiempo de simulación usando este método de truncamiento es muy lento. Por lo que, si un usuario desea simular un análisis monoestático utilizando el método IIEE para obtener un resultados muy precisos, la mejor opción es utilizar un clúster HPC que le permita reducir el tiempo de simulación realizando la simulación en paralelo.



**Figura 3.41:** Directividad de la bocina para  $\Phi = 0$  usando ABC



**Figura 3.42:** Directividad de la bocina para  $\Phi = 90$  usando ABC

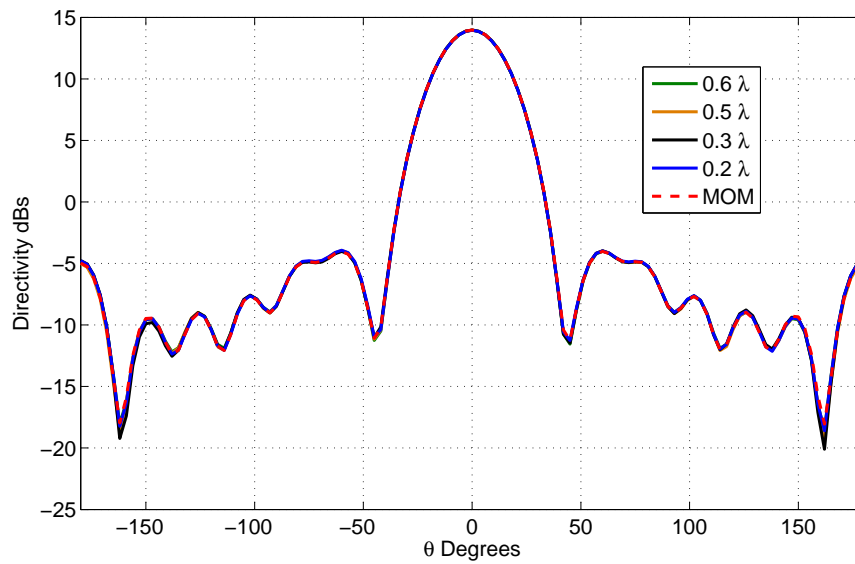


**Figura 3.43:** Directividad de la bocina para  $\Phi = 0$  usando IIEE

### 3.3. PROBLEMAS DE RADIACIÓN/SCATTERING

**Tabla 3.15:** Comparativa Memoria tiempo de cómputo

| Método | Distancia    | Elementos | Incógnitas | Tiempo (s) |
|--------|--------------|-----------|------------|------------|
| ABC    | $0.2\lambda$ | 36209     | 232148     | 76.36      |
| ABC    | $0.3\lambda$ | 39399     | 253140     | 95.64      |
| ABC    | $0.5\lambda$ | 42640     | 274430     | 125.46     |
| ABC    | $0.6\lambda$ | 46637     | 300528     | 234.36     |
| IEE    | $0.2\lambda$ | 36209     | 232148     | 144.04     |
| IEE    | $0.3\lambda$ | 39399     | 253140     | 182.68     |
| IEE    | $0.5\lambda$ | 42640     | 274430     | 234.22     |
| IEE    | $0.6\lambda$ | 46637     | 300528     | 614.80     |



**Figura 3.44:** Directividad de la bocina para  $\Phi = 90$  usando IEE



---

---

# CAPÍTULO 4

---

## CONCLUSIONES

### 4.1 Conclusiones finales y posibles líneas de investigación

Tal y como se ha visto en el Capítulo 2, uno de los aspectos principales de este PFC, consiste en mejorar la interfaz gráfica de usuario del código de simulación electromagnética existente en el grupo de investigación con el que colaboro. En dicho capítulo se ha mostrado como desarrollar un asistente para crear el modelado de algunas estructuras complejas utilizando como base el software de propósito general GID. De este modo, y gracias a la codificación de algunos archivos en el mismo lenguaje en el que está escrito GID, la creación de algunas estructuras, como guías de onda o cables coaxiales es posible únicamente introduciendo sus parámetros característicos. Además, en este Capítulo 2 se ha detallado como implementar el asistente desde cero, indicando primero los paquetes utilizados y los métodos que se encuentran definidos previamente, para posteriormente indicar los implementados por nosotros mismos.

Tal y como se ha desarrollado este asistente, queda abierto a la introducción de ciertas mejoras posteriores, como por ejemplo dotar de funcionalidad al botón «pick», que permitiría la obtención del vector de posicionamiento a partir de uno ya existente. También está abierto a la introducción de otras estructuras de una forma sencilla. En esta versión del asistente, el lector ya puede hacerse una idea de las posibilidades que ofrece GID para su customización, y de los beneficios que esto conlleva en términos de ahorro de tiempo.

#### 4.1. CONCLUSIONES FINALES Y POSIBLES LINEAS DE INVESTIGACIÓN

---

Como se comentó en el Capítulo 1, para que cualquier software sea confiable, es necesario realizar una validación/verificación del mismo. Es por esto que en el Capítulo 3 se ha realizado un análisis de múltiples estructuras para comprobar los resultados obtenidos utilizando el código de simulación electromagnética que posee el grupo de investigación.

A la vista del análisis de estructuras cerradas realizado con 10 geometrías diferentes, se ha concluido que el resultado de las simulaciones van a depender en gran medida del mallado utilizado. Así pues para mallados muy pequeños (del entorno de  $0.08\lambda$  de tamaño), se obtienen unos resultados muy precisos. Del mismo modo, por debajo de  $0.1\lambda$ , el tiempo de cómputo crece de forma drástica, por lo que salvo que queramos obtener unos resultados muy precisos para algún caso concreto, realizar simulaciones con mallados en torno a  $0.1\lambda$  de tamaño nos va a proporcionar una buena relación entre la precisión obtenida y el tiempo de cómputo necesario.

En el caso de los problemas de radiación, tras analizar 3 estructuras utilizando 2 métodos de truncamiento de malla diferentes, vemos como el uso de la condición ABC de primer orden viene condicionado fuertemente por la distancia de la frontera al objeto. Para conseguir resultados precisos es necesario colocar la frontera a distancias mayores de  $0.45\lambda$ , esto hace que el tamaño del problema sea muy grande haciendo que aumente el tiempo de simulación. En cambio al utilizar el método IEEE, los resultados no se ven afectados por la distancia de la frontera al objeto. Esta distancia en lo que repercute es en el número de iteraciones necesarias para conseguir los resultados y en el tiempo empleado en cada una de ellas.

Para problemas con un número de excitaciones pequeño, como en problemas de antenas o análisis biestático de objetos, obtenemos un resultado óptimo al colocar la frontera en el entorno de  $0.2$  y  $0.1\lambda$ . Por el contrario al analizar un análisis RCS monoestático, el tiempo necesario es muy elevado debido al gran número de excitaciones presentes en el problema, por lo que para conseguir unos resultados precisos, la mejor opción es utilizar un clúster HPC que permita reducir el tiempo de simulación realizando la simulación en paralelo.

Por último, este PFC presenta un Anexo que comprende un Manual de Usuario (Ver Anexo A). Este manual pretende ser una guía de utilización del asistente desarrollado. De tal forma que, además de explicar su utilización de forma sencilla a un usuario no experto, también muestra las ventajas de su uso, ya que como se puede comprobar, facilita la definición de los puertos de excitación, reduciendo sensiblemente el tiempo destinado al modelado.

---

# CAPÍTULO 5

---

## PRESUPUESTO

En este capítulo se muestran los costes globales de realización del presente Proyecto Final de Carrera. Estos costes están asociados principalmente a gastos materiales y de personal. En primer lugar se realiza un cómputo de las horas empleadas en la realización de cada una de las fases del siguiente PFC sumando tanto las horas de diseño, ejecución y redacción de la memoria, tal como se indica en la tabla 5.1. Haciendo el cómputo global de las horas empleadas en las distintas fases del proyecto, suma un total de 1260 horas. Teniendo en cuenta que el salario de un ingeniero superior de telecomunicaciones junior ronda los 27 €/hora, el coste de personal para este proyecto sería de 34020 €. A esta cantidad hay que añadirle el salario del tutor del proyecto, si estimamos un salario de ingeniero senior de 35 €/hora, y una dedicación de unas 120 horas en tareas de diseño y seguimiento del proyecto, nos daría un coste adicional de 4200 €. Tal como se indica en la tabla 5.2 El coste de personal asociado a este proyecto sería de 38220 €

Respecto a los costes materiales del proyecto, el principal es la adquisición de un ordenador de sobremesa de nivel medio. En lo que se refiere a software, la herramienta principal para realizar las simulaciones es el pre/postprocesador GID cuyo coste es de 665 €, el resto de las herramientas utilizadas son a coste cero ya que han sido desarrolladas dentro del propio grupo de investigación. Estos gastos vienen especificados en la tabla 5.3. El cálculo de la amortización de estos bienes, se calculan según la fórmula  $A/(BCD)$  donde **A** es el número de meses desde la fecha de facturación en que el equipo es utilizado, **B** es el periodo de depreciación (60 meses), **C** es el coste del equipo y **D** es el % de uso dedicado al proyecto, en este caso 100 %. Los costes indirectos se han calculado como el 20 % de los gastos de personal, equipos y software, resultando un total 7729.2€, según se muestra en la tabla 5.4. Por lo que los costes totales asociados a este proyecto, son los reflejados en la tabla 5.5.

**Tabla 5.1:** Fases del Proyecto

| Fase                       | Duración (Horas) |
|----------------------------|------------------|
| Diseño y Planificación     | 80               |
| Implementación Herramienta | 320              |
| Evaluación Herramienta     | 60               |
| Análisis y Verificación    | 480              |
| Redacción Memoria          | 320              |
| <b>Total</b>               | <b>1260</b>      |

**Tabla 5.2:** Coste personal

| Categoría          | Dedicación | Coste Hora   | Coste Total  |
|--------------------|------------|--------------|--------------|
| Ingeniero Superior | 1260       | 27           | 34020        |
| Jefe de Proyecto   | 120        | 35           | 4200         |
|                    |            | <b>Total</b> | <b>38220</b> |

**Tabla 5.3:** Costes materiales

| Nombre             | Coste | Dedicación | Periodo depreciación | Coste      |
|--------------------|-------|------------|----------------------|------------|
| Ordenador Personal | 800   | 12 (Meses) | 60 (Meses)           | 160        |
| GID                | 665   | 12 (Meses) | 30 (Meses)           | 266        |
|                    |       |            | <b>Total</b>         | <b>426</b> |

**Tabla 5.4:** Costes indirectos

| Descripción                    | Coste         |
|--------------------------------|---------------|
| Costes Personales              | 38220         |
| Costes Materiales              | 426           |
| <b>Costes indirectos (20%)</b> | <b>7729.2</b> |

**Tabla 5.5:** Costes Totales

| Descripción       | Coste          |
|-------------------|----------------|
| Costes Personales | 38220          |
| Costes Materiales | 426            |
| Costes Indirectos | 7729.2         |
| <b>Total</b>      | <b>46375.2</b> |

---

---

# APÉNDICE A

---

## MANUAL DE USUARIO

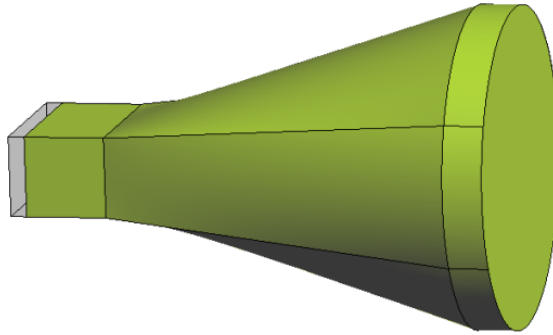
En este apéndice se muestra un manual de usuario, gracias al cual, se posibilita que cualquier persona sin unos conocimientos técnicos avanzados, pueda utilizar la interfaz gráfica de usuario con la que se ha trabajado en este PFC, incluyendo una explicación detallada del funcionamiento y el uso del asistente desarrollado.

En este manual, se indica la forma de iniciar el asistente, se muestran las opciones de configuración del mismo y se ofrece una guía sencilla con la creación de la geometría completa de dos ejemplos. Estas estructuras son una antena tipo bocina y una tipo parche, ya que estas antenas tienen unas excitaciones que se pueden generar utilizando el asistente implementado en este PFC.

### A.1 Bocina

El primer ejemplo de este manual consiste en la creación de una bocina circular en banda X. Una definición sencilla de una antena de este tipo, sería una antena que consiste en una guía de onda en la cual el área de la sección se va incrementando progresivamente hasta un extremo abierto, que se comporta como una apertura. Un ejemplo de la bocina usada en esta sección se muestra en la figura A.1.

El primer paso a la hora de realizar la simulación de la antena consiste en la creación de la geometría de la bocina. Para ello se irán dibujando las diferentes formas necesarias

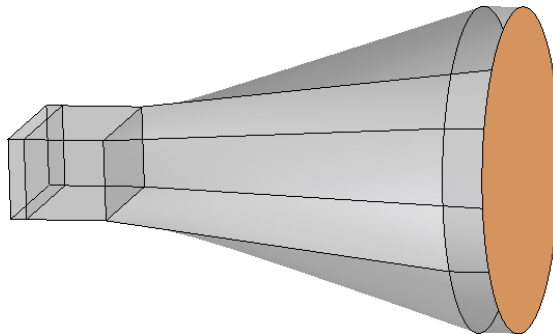


**Figura A.1:** Bocina

para su construcción, utilizando las herramientas que proporciona la interfaz de usuario y, finalmente, proporcionando la excitación adecuada con la ayuda del asistente.

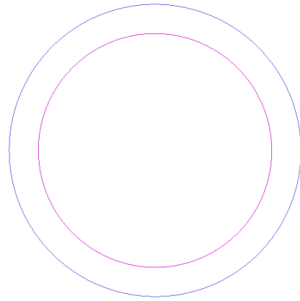
### A.1.1 Creación de la geometría

Para la creación de una bocina como la vista en la figura A.1, en primer lugar se va a definir la boca de la apertura de la bocina. Por lo que el primer paso será definir una circunferencia cuyo radio sea la mitad de la apertura deseada. La circunferencia que se va a crear corresponderá a la mostrada en la figura A.2, y se realizará de la forma que se indica a continuación.



**Figura A.2:** Boca de la Apertura

- Selección de las unidades
  1. En la esquina superior derecha de la interfaz seleccionar **mm** como unidades
- Creación de la circunferencia 1
  1. Seleccionar la opción **Geometry** → **Create** → **Object** → **Circle**
  2. Introducir las coordenadas del centro, en este caso es el origen (0,0,0)
  3. Introducir el valor de la normal, en este caso seleccionaremos **PositiveZ**
  4. Introducir el valor del radio de la circunferencia, en este caso es de 25.4 mm

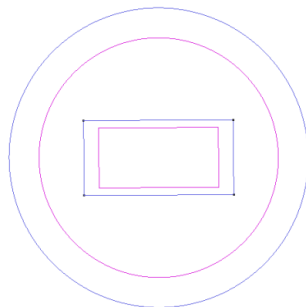


**Figura A.3:** Circunferencia

5. Pulsar la tecla ESC para finalizar la creación del círculo. La figura A.3 muestra la circunferencia de la interfaz gráfica.

A continuación, se crea un rectángulo auxiliar con las mismas dimensiones que la boca de la guía rectangular, que nos permitirá posteriormente definir los puntos de unión de la circunferencia a la guía rectangular.

- Creación del rectángulo auxiliar:
  1. Seleccionar la opción **Geometry** → **Create** → **Object** → **Rectangle**
  2. Introducir las coordenadas de una de las esquinas, como queremos que esté centrado en el origen, el valor de este punto será  $(-12.7, -6.35, 0)$ , correspondiente a la mitad del ancho y alto de la guía.
  3. Introducir las coordenadas de la esquina contraria, el valor de este punto será  $(12.7, 6.35, 0)$
  4. Pulsar la tecla ESC para finalizar la creación del rectángulo (ver figura A.4).

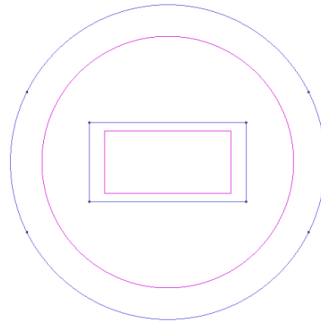


**Figura A.4:** Circunferencia con rectángulo auxiliar

Con ayuda del rectángulo auxiliar creado, se van a definir los puntos desde los que partirán las líneas que engancharán con la guía, para ello hay que realizar lo siguiente:

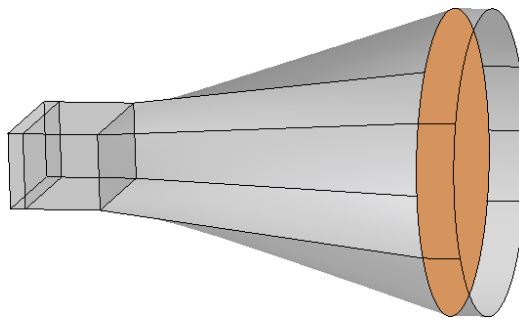
- Definición de los puntos de enganche:
  1. Seleccionar la opción **Geometry** → **Edit** → **Divide** → **Lines** → **Near Point**
  2. A continuación seleccionar el vértice superior izquierdo del rectángulo.
  3. Seleccionar la línea que conforma la circunferencia
  4. Pulsar la tecla escape para finalizar.
  5. Repetiremos estos pasos para los otros 3 vértices restantes.

En este momento ya se tiene la circunferencia que conforma la parte exterior de la apertura de la bocina con los 4 puntos auxiliares desde donde partirán las líneas que se unirán con la boca de la guía como muestra la figura A.5.



**Figura A.5:** Circunferencia con los puntos de unión

A continuación hay que dibujar una circunferencia concéntrica a la anterior, separada una distancia  $l$ , estas dos circunferencias constituyen la apertura de la bocina. La circunferencia que se va a crear corresponderá a la mostrada en la figura A.6.

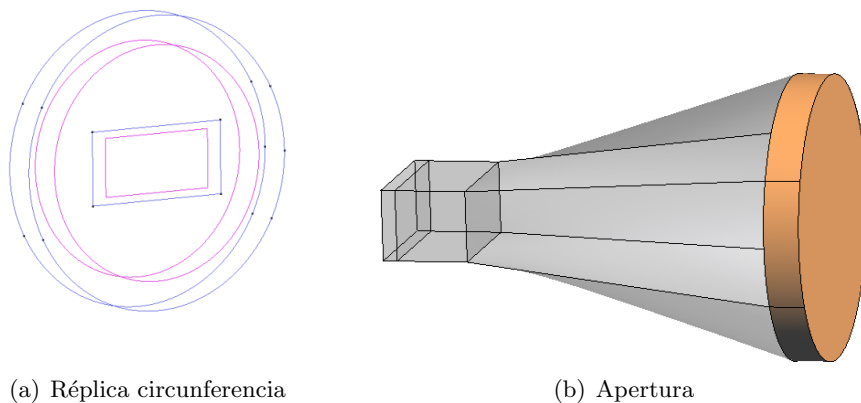


**Figura A.6:** Boca Apertura 2



- Creación de la circunferencia 2
  1. Para ello vamos a realizar una copia de la circunferencia que tenemos creada, ya que ya tiene los puntos de corte definidos.
  2. Seleccionar la opción **Utilities** → **Copy**
  3. Entre las opciones marcaremos **Entities type** → **All types** ; **Transformation** → **Translation** ; **Duplicate Entities** → **NO**
  4. La copia se hará desde el punto origen (0,0,0) al punto destino (0,0,-6.35)
  5. Pulsar la opción «Select» y seleccionar la superficie del círculo ya creado.
  6. Pulsar la tecla escape para finalizar la creación del círculo.

En este momento ya se tiene la apertura de la bocina perfectamente definida, incluyendo los puntos desde los que partirán las líneas que se unen a la boca de la guía, tal como puede verse en la figura A.7.

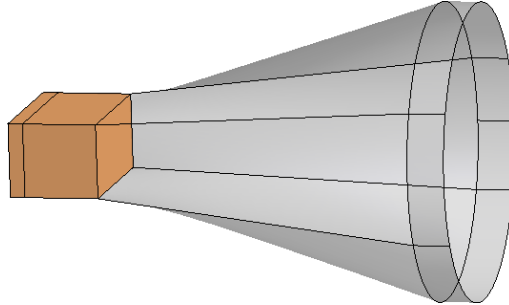


**Figura A.7:** Apertura de la Bocina

El siguiente paso es introducir la guía rectangular, recordemos tal como se dijo al inicio de este manual, que dicha guía irá creciendo en su sección hasta conformar la apertura de la bocina. Se puede ver la guía en la figura A.8.

- Introducción de la guía rectangular.
  1. A continuación vamos a introducir la guía rectangular, pero en lugar de dibujarlo manualmente, utilizaremos el wizard desarrollado. Para poder dar comienzo al asistente, seleccionaremos el icono con forma de varita mágica de la parte superior derecha de la barra de herramientas.
  2. Una vez iniciado el asistente, seleccionar la opción «Waveguide» y pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso.

## A.1. BOCINA



**Figura A.8:** Guía Rectangular

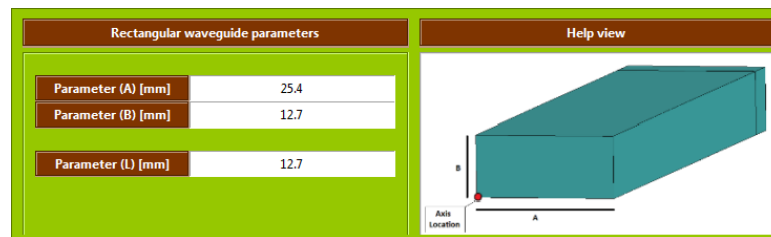
- Introducir el valor del punto donde queremos situar la guía, en este caso será  $(-12.7, 6.35, -63.5)$ , que corresponde a la esquina superior izquierda de la apertura de la guía, este punto coincide con su análogo del rectángulo auxiliar creado anteriormente desplazado en el eje Z (ver figura A.9).

| Axis location   | Puerta horizontal | Puerta vertical |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
|---|-------------------|-----------------|--------|------|--------|-------|---|-------------|-----|-------------|-----|-------------|-----|-------------------|--------|---|-------------|-----|-------------|------|-------------|-----|-------------------|---------|
| <table border="1"><tr><td>X [mm]</td><td>-12.7</td></tr><tr><td>Y [mm]</td><td>6.35</td></tr><tr><td>Z [mm]</td><td>-63.5</td></tr></table> | X [mm]            | -12.7           | Y [mm] | 6.35 | Z [mm] | -63.5 | <table border="1"><tr><td>X-component</td><td>1.0</td></tr><tr><td>Y-component</td><td>0.0</td></tr><tr><td>Z-component</td><td>0.0</td></tr><tr><td>Predefine vectors</td><td>X-axis</td></tr></table> | X-component | 1.0 | Y-component | 0.0 | Z-component | 0.0 | Predefine vectors | X-axis | <table border="1"><tr><td>X-component</td><td>0.0</td></tr><tr><td>Y-component</td><td>-1.0</td></tr><tr><td>Z-component</td><td>0.0</td></tr><tr><td>Predefine vectors</td><td>-Y-axis</td></tr></table> | X-component | 0.0 | Y-component | -1.0 | Z-component | 0.0 | Predefine vectors | -Y-axis |
| X [mm]  | -12.7             |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Y [mm]  | 6.35              |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Z [mm]  | -63.5             |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| X-component   | 1.0               |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Y-component   | 0.0               |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Z-component   | 0.0               |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Predefine vectors   | X-axis            |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| X-component   | 0.0               |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Y-component   | -1.0              |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Z-component   | 0.0               |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |
| Predefine vectors   | -Y-axis           |                 |        |      |        |       |   |             |     |             |     |             |     |                   |        |   |             |     |             |      |             |     |                   |         |

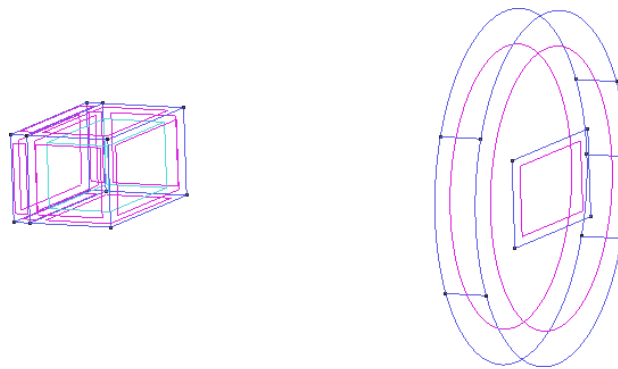
**Figura A.9:** Localización Guía Rectangular

- Indicar las direcciones de los ejes Horizontal y Vertical que conforman la puerta de la guía. En este caso seleccionaremos las opciones «X-Axis» e «-Y-Axis» respectivamente de cada uno de los desplegables. A continuación pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso.
- Introducir las dimensiones de la guía. En este caso serán 25.4 mm para el parámetro A y 12.7 mm para el parámetro B. Introducir el valor de la longitud de la guía, en este caso 12.7 mm (ver figura A.10).
- Pulsar el botón «Finish» situado en la parte inferior del asistente para finalizar la creación de la guía rectangular.

Como puede verse en la figura A.11, ya se tiene perfectamente dibujada la apertura de la bocina por un lado, y por otro la guía rectangular. A continuación hay que definir las paredes laterales que conforman la apertura de la bocina (ver figura A.12).

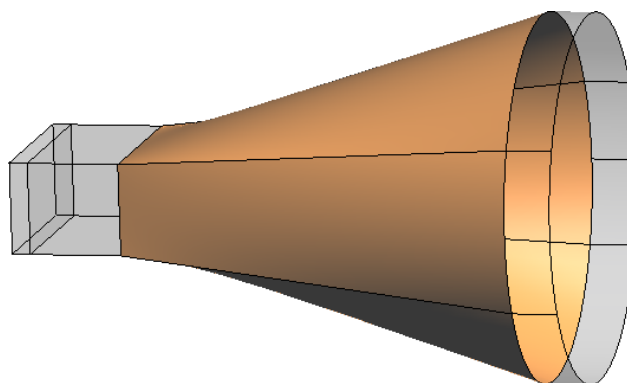


**Figura A.10:** Parámetros Guía Rectangular

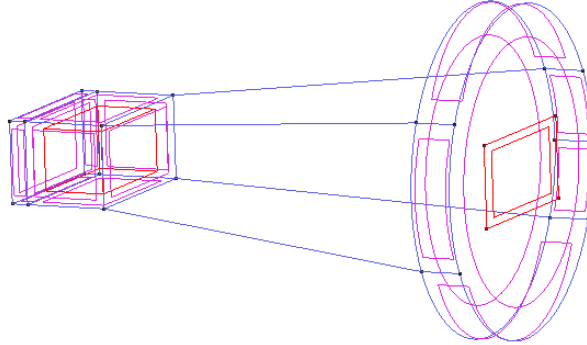


**Figura A.11:** Guía + Apertura

- Creación de las líneas de contorno
  1. En primer lugar, seleccionar la opción **Geometry** → **Create** → **Straight line**. Seleccionaremos uno de los vértices de la apertura de la guía, y por otro lado, el punto más próximo perteneciente a la circunferencia. En este caso uniremos los puntos (15,11), (16,12), (17,14) y (18,13).
  2. Para no crear nuevos puntos y utilizar los ya existentes, hay que seleccionar la opción **Join**. De este modo no se duplicarán entidades.
  3. Tras seleccionar los dos puntos hay que pulsar la tecla escape para finalizar la creación de la línea.



**Figura A.12:** Superficies laterales

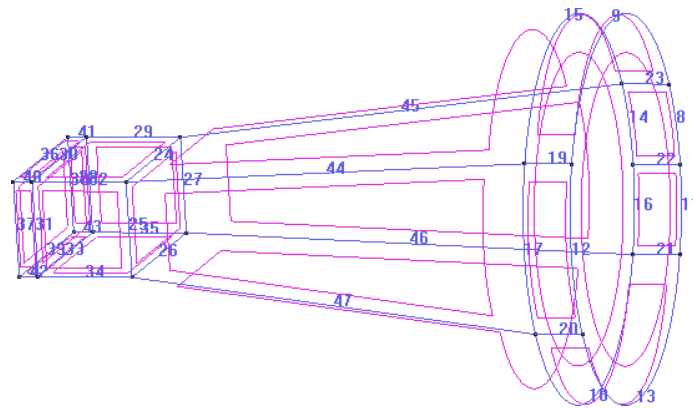


**Figura A.13:** Eliminar entidades auxiliares

4. Pulsar la tecla escape para finalizar la creación de las líneas.

A continuación se procede a eliminar las entidades auxiliares que no forman parte de la bocina. Las entidades que se van a eliminar, pueden verse en la figura A.13.

- Eliminación de entidades auxiliares
  1. Seleccionaremos la opción **Geometry** → **Delete** → **All types**. Seleccionaremos la superficie, y las cuatro líneas y puntos que conforman el rectángulo auxiliar creado al principio.
  2. Eliminar el volumen generado al crear la guía mediante el asistente.
  3. Pulsar la tecla escape para finalizar la eliminación de entidades.
- Creación de las superficies
  1. En primer lugar seleccionaremos la opción **View** → **Label** → **All in** → **Lines**. De esta forma aparecerán en pantalla las etiquetas de las líneas.
  2. Para crear las superficies seleccionaremos la opción **Geometry** → **Create** → **NURBS surface** → **By contour**
  3. Primero crearemos el contorno lateral de la apertura de la bocina, correspondiente a la parte lateral de los dos círculos concéntricos. Seleccionar las líneas que conforman las superficies. En este caso, la primera superficie vendrá determinada por las líneas (9, 15, 19 y 23). A continuación presionar escape para terminar la creación de la superficie.
  4. Repetir el proceso para crear las otras 4 superficies, seleccionando las líneas (12, 17, 19 y 20), (13, 18, 20 y 21), (11, 16, 21 y 22) y (8, 14, 22 y 23).



**Figura A.14:** Superficies Laterales

5. A continuación crearemos las cuatro superficies que conforman las paredes laterales de la apertura de la bocina, para ello repetiremos el proceso anterior seleccionando las líneas (15, 24, 44 y 45), (17, 25, 44 y 47), (18, 26, 46 y 47) y (14, 16, 27, 45 y 46).
6. Pulsar la tecla escape para finalizar la creación de las superficies.

Como puede verse en la figura A.14, ya se tiene la estructura de la bocina perfectamente dibujada, con todas las superficies que la conforman. El último paso es darle volumen a la misma.

- Creación del volumen

1. En primer lugar seleccionaremos la opción **View** → **Label** → **All in** → **Surfaces** De esta forma aparecerán en pantalla las etiquetas de las Superficies.
2. Para crear las superficies seleccionaremos la opción **Geometry** → **Create** → **Volume** → **By contour**
3. Seleccionaremos las superficies que conforman el contorno deseado, en este caso son (9, 10, 11, 12, 14, 25, 26, 27, 28, 20, 21, 22, 23, 24 y 1)
4. Pulsar la tecla escape para finalizar la creación del volumen.

La figura A.15 muestra la bocina desarrollada siguiendo los pasos anteriores.

De esta forma concluye la creación de la bocina circular en banda X. Como se ha podido observar, la interfaz de usuario utilizada, proporciona todas las herramientas necesarias para crear esta y otras estructuras. En este ejemplo, se presenta una de las múltiples formas de crear una bocina de estas características. En este caso se ha optado por generar en primer lugar la apertura de la bocina, a continuación se ha introducido la guía rectangular que proporcionará la excitación mediante el uso del asistente desarrollado

en este PFC. Finalmente, se han generado las formas necesarias para crear las paredes de la apertura de la bocina usando de nuevo la interfaz de usuario. Por último se dota a esta estructura del volumen concluyendo de este modo su creación.

## A.2 Parche

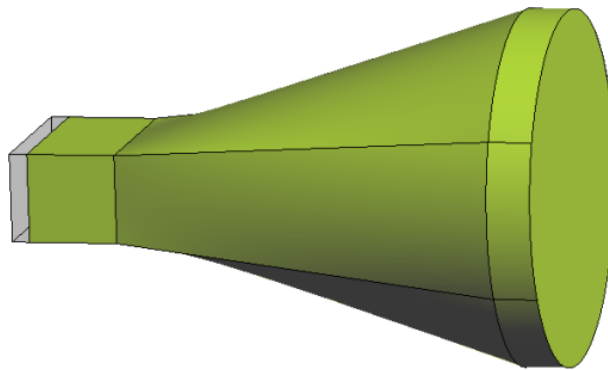
Las antenas impresas, y concretamente las antenas de parche, cobran cada día mayor protagonismo en diversos sistemas de comunicaciones, debido a las importantes ventajas que ofrecen frente a otros tipos de antena, como su versatilidad o su bajo coste. Una antena de parche consta principalmente de un plano de masa, una capa de dieléctrico, un sustrato y una metalización. Debe incluir también un mecanismo de excitación. En nuestro caso la metalización será de tipo circular, y la excitación se realizará mediante un cable coaxial. Un ejemplo de este tipo de antenas se muestra en la figura A.16.

### A.2.1 Creación de la geometría

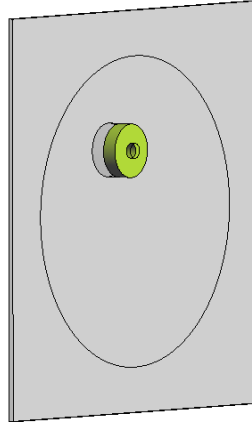
A modo de resumen, y tal como puede verse en la figura A.16, la geometría consistirá en una circunferencia que irá encuadrada en una caja de espesor delgado. Posteriormente, un cable coaxial conectará con la circunferencia por medio del vivo y con la caja por medio de la camisa.

El primer paso será definir una circunferencia cuyo radio sea la mitad de la apertura deseada. La circunferencia que se va a crear corresponderá a la mostrada en la figura A.17, y se realizará de la forma que se indica a continuación.

- Selección de las unidades
  1. En la esquina superior derecha de la interfaz seleccionar **mm** como unidades
- Creación de la circunferencia



**Figura A.15:** Geometría de la bocina circular



**Figura A.16:** Parche

1. Seleccionar la opción **Geometry** → **Create** → **Object** → **Circle**
2. Introducir las coordenadas del centro, en este caso es el origen  $(0,0,0)$
3. Introducir el valor de la normal, en este caso seleccionaremos **PositiveZ**
4. Introducir el valor del radio de la circunferencia, en este caso es de 23.2
5. Pulsar la tecla escape para finalizar la creación del círculo.

El siguiente paso es crear una caja que encuadre al círculo definido anteriormente. Para ello definiremos dos rectángulos concéntricos con el círculo, separados una distancia  $l$ , esta distancia será el espesor de la caja.

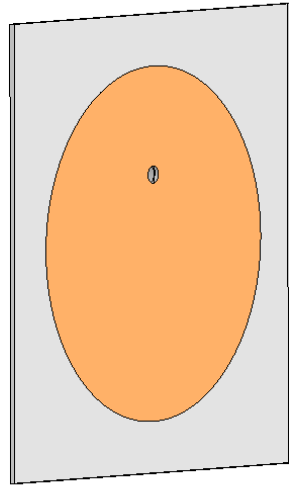
- Creación de la caja

1. Seleccionar la opción **Geometry** → **Create** → **Object** → **Rectangle**
2. Introducir las coordenadas de una de las esquinas, en este caso  $(30,-30,0)$ .
3. Introducir las coordenadas de la esquina contraria, en este caso  $(-30,30,0)$ .
4. Pulsar la tecla escape para finalizar la creación del rectángulo.

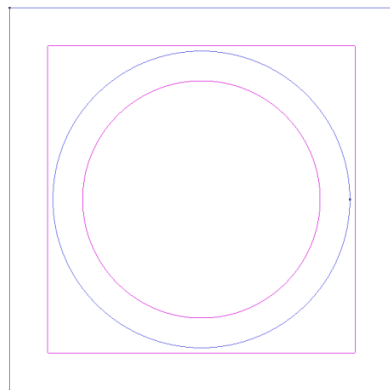
El rectángulo creado se corresponde con el mostrado en la figura A.18(b). Tal como puede verse en la figura A.18(a), se tiene una circunferencia y un rectángulo concéntricos.

A continuación se crea un segundo rectángulo de las mismas dimensiones a una distancia  $l$  igual al espesor elegido, en este caso 0.7 mm.

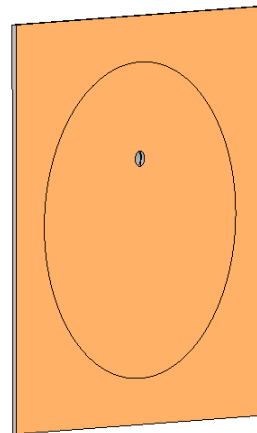
1. Introducir las coordenadas de una de las esquinas, en este caso  $(30,-30,-0.7)$ .



**Figura A.17:** Parche-Circunferencia



(a) Circunferencia + Rectángulo

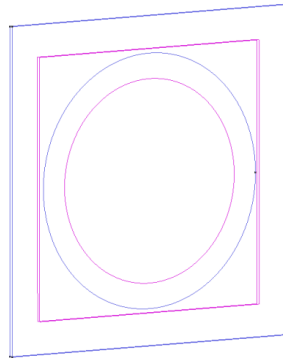


(b) Rectángulo

**Figura A.18:** Caja Paso 1

2. Introducir las coordenadas de la esquina contraria, en este caso  $(-30,30,-0.7)$ .
3. Pulsar la tecla escape para finalizar la creación del rectángulo.
4. A continuación uniremos los vértices con líneas para cerrar el contorno. Seleccionar la opción **Geometry** → **Create** → **Straight line**. Seleccionaremos una de las esquinas y su punto más próximo. En este caso uniremos los puntos (2-6), (3-7), (4-8) y (5-9). Habrá que seleccionar la opción **Join** para selec-





**Figura A.19:** Circunferencia + Dos Rectángulos

cionar los puntos existentes. Tras seleccionar los dos puntos hay que pulsar la tecla escape para finalizar la creación de la línea.

5. Pulsar la tecla escape para finalizar la creación de las líneas.

En este punto se tiene el contorno de la caja perfectamente dibujado tal como muestra la figura A.19, a continuación se definen las superficies laterales de la caja.

- Creación de las superficies
  1. En primer lugar seleccionaremos la opción **View** → **Label** → **All in** → **Lines**. De esta forma aparecerán en pantalla las etiquetas de las líneas.
  2. Para crear las superficies seleccionaremos la opción **Geometry** → **Create** → **NURBS surface** → **By contour**
  3. Seleccionar las 4 líneas que conforman las superficies. En este caso, la primera superficie vendrá determinada por las líneas (5, 9, 10 y 11). A continuación presionar escape para terminar la creación de la superficie.
  4. Repetir el proceso para crear las otras 3 superficies, seleccionando las líneas (4, 8, 10 y 13), (3, 7, 12 y 13) y (2, 6, 11 y 12).

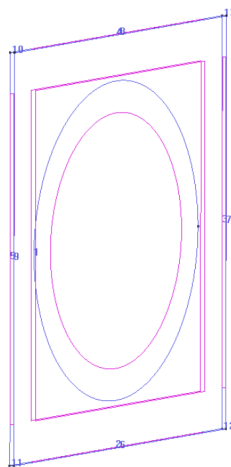
Tal como muestra la figura A.20, las superficies laterales que forman la caja están perfectamente definidas. A continuación eliminaremos las superficies generadas al crear el círculo y los rectángulos para posteriormente crear otras nuevas de la manera que se explica.

1. Para borrar las superficies lo haremos mediante **Geometry** → **Delete** → **Surfaces**, seleccionaremos las superficies 1, 2 y 3 y pulsaremos la tecla escape para finalizar la selección (ver figura A.21)
2. Crearemos una superficie que comprenderá la parte del rectángulo que no coincide con la circunferencia, para ello seleccionaremos la opción **Surface** mediante **Geometry** → **Create** → **NURBS surface** → **By contour** y seleccionaremos las líneas (1, 2, 3, 4 y 5)

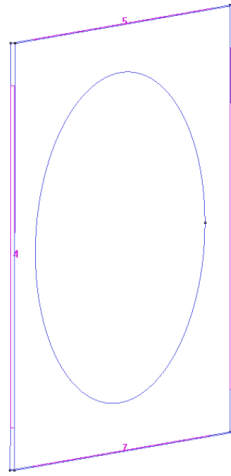
3. Pulsar la tecla escape para finalizar la creación de las superficies.

De este modo ya se tiene la caja perfectamente definida (ver figura A.22).

- Introducción del cable coaxial.
  1. A continuación vamos a introducir el cable coaxial, pero en lugar de dibujarlo manualmente, utilizaremos el wizard desarrollado. Para ello seleccionaremos el icono con forma de varita mágica de la parte superior derecha de la pantalla, de forma análoga a como se hizo al crear la bocina (ver figura A.23)
  2. Seleccionar la opción «Coaxial Cable» y pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso.
  3. Introducir el valor del punto donde queremos situar el cable coaxial, en este caso será (0,9,0), corresponde al punto origen donde pintamos la primera circunferencia desplazado en el eje Y.
  4. Indicar el eje longitudinal del cable coaxial, en este caso seleccionaremos la opción «-Z-Axis» del desplegable, tal como se ve en la figura (ver figura A.24). A continuación pulsar el botón «Next» situado en la parte inferior del asistente para avanzar al siguiente paso.
  5. Introducir los valores de los radios interior y exterior del cable coaxial. En este caso serán 1.12 mm para el radio del vivo y 4 mm para el radio de la camisa.
  6. Introducir los valores de las longitudes. La mayor será de 2.8 mm y la menor de 2.1 mm, la diferencia se corresponde con el espesor del dieléctrico. Estos parámetros se introducen según se muestra en la figura A.25.
  7. Pulsar el botón «Finish» situado en la parte inferior del asistente para finalizar la creación del cable coaxial. Véanse las figuras A.26(a) y A.26(b).

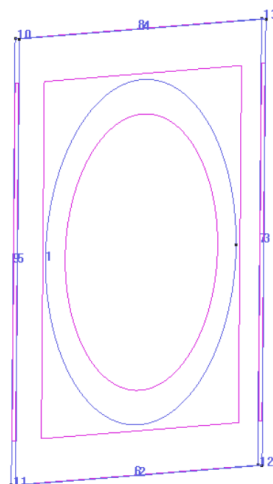


**Figura A.20:** Circunferencia + Caja



**Figura A.21:** Eliminación Superficies

- Creación de las superficies 2
  1. En primer lugar seleccionaremos la opción **View** → **Label** → **All in** → **Lines**. De esta forma aparecerán en pantalla las etiquetas de las líneas.
  2. Para crear las superficies seleccionaremos la opción **Geometry** → **Create** → **NURBS surface** → **By contour**
  3. Crearemos una superficie que comprenderá el cuadrado menos la parte del coaxial de radio mayor (camisa). Seleccionar las 6 líneas que conforman la supercie. En este caso, vendrá determinada por las líneas (6, 7, 8, 9, 22 y 24). A continuación presionar escape para terminar la creación de la superficie.

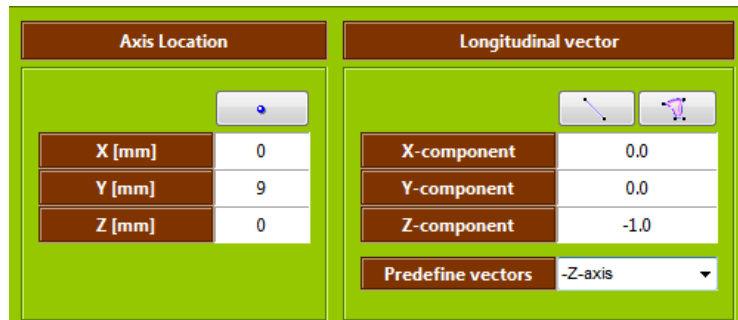


**Figura A.22:** Caja

## A.2. PARCHE



**Figura A.23:** Inicio del asistente.



**Figura A.24:** Localización del Parche

4. Crearemos una superficie que comprenderá el parche menos la parte del coaxial de radio menor (vivo) y vendrá determinada por las líneas (1, 26 y 27). A continuación presionar escape para terminar el proceso.
5. Crearemos una superficie que será la base de la punta del vivo, determinada por las líneas (26 y 27). A continuación presionar escape para terminar la creación de la superficie y otra vez para finalizar la creación de las superficies.

De esta forma concluye la creación del parche, tal y como se puede ver en las figuras A.27(a) y A.27(b) que muestran el parche completo.

Para llevar a cabo este proceso, se ha comenzado por dibujar la metalización que en este caso ha sido de forma circular. Posteriormente se ha encuadrado en una caja que representa el sustrato. Para todo esto, se ha utilizado la interfaz de usuario que ofrece las herramientas necesarias para crear círculos, polígonos, líneas, así como eliminar o crear nuevas superficies hasta obtener la geometría deseada. Posteriormente se ha hecho uso del asistente desarrollado en este PFC para proporcionar la excitación. En este caso, se ha hecho mediante un cable coaxial. Finalmente y utilizando de nuevo las herramientas de la interfaz de usuario, se han generado las superficies necesarias para insertar el cable coaxial en el parche.

De este modo, se han ilustrado dos ejemplos de como realizar antenas de uso común mediante el uso de la interfaz de usuario, añadiendo la excitación por medio del asistente desarrollado en este PFC. Utilizando las dos variantes que ofrece, siendo estas una guía rectangular y un cable coaxial.

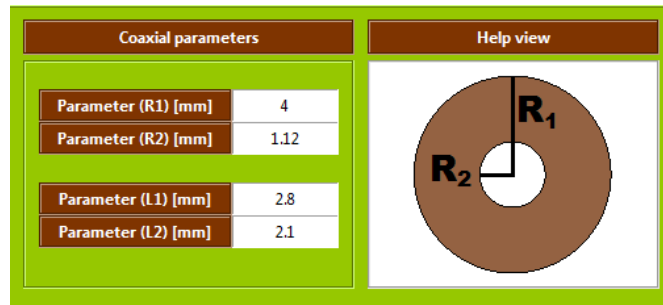
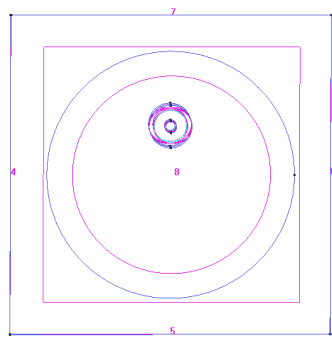
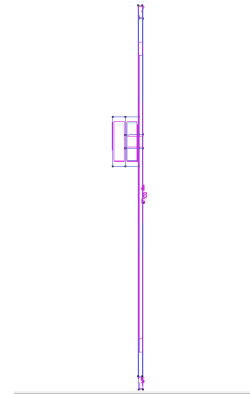


Figura A.25: Parámetros del Parche

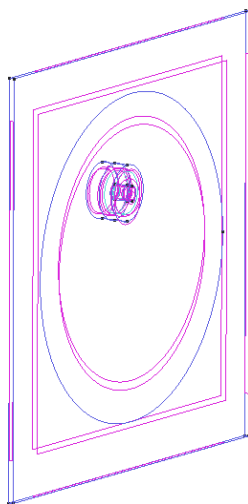


(a) Lateral

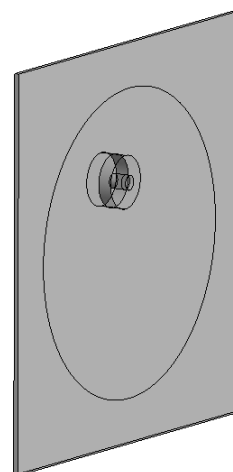


(b) Frontal

Figura A.26: Parche



(a) Parche1



(b) Parche2

Figura A.27: Parche



---

# BIBLIOGRAFÍA

- [1] R. F. Harrington, *Field Computation by Moment Method*. The Macmillan company, 1968.
- [2] Y. Zhang and T. K. Sarkar, *Parallel Solution of Integral Equation Based EM Problems in the Frequency Domain*. Wiley-IEEE Press, July 2009.
- [3] Y. Zhang, T. K. Sarkar, X. Zhao, D. Garcia-Donoro, W. Zhao, M. Salazar, and S. Ting, *Higher Order Basis Based Integral Equation Solver (HOBBIES)*. John Wiley & Sons, Inc., 2012. ISBN: 9781118140659.
- [4] J. M. Jin, *The Finite Element Method in Electromagnetics*. John Wiley & Sons, Inc., 1993.
- [5] M. Salazar-Palma, T. K. Sarkar, L.-E. Garcia-Castillo, T. Roy, and A. Djordjevic, *Iterative and Self-Adaptive FiniteElements in Electromagnetic Modeling*. Norwood, MA: Artech House Publishers, 1998.
- [6] J. L. Volakis, A. Chatterjee, and L. C. Kempel, *Finite Element Method for Electromagnetics*. IEEE/OUP Series on Electromagnetic Wave Theory, IEEE Press, 1998.
- [7] K. S. Yee, “Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media,” *IEEE Transaction on Antennas and Propagation*, vol. 14, pp. 302–307, May 1966.
- [8] K. L. Shlager and J. B. Schneider, “A selective survey of the finite-difference time-domain literature,” *IEEE Antennas and Propagation Magazine*, vol. 37, pp. 39–56, Aug 1995.
- [9] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Norwood, MA: Artech House Publishers, Inc., 2005.

- [10] A. Carnicero, “Introducción al método de los elementos finitos,” tech. rep., Universidad Pontificia Comillas, 2008.
- [11] D. García-Donoro, I. Martinez-Fernandez, L. E. García-Castillo, and M. Salazar-Palma, “Hofem: A higher order finite element method electromagnetic simulator,” in *International Conference on Computational Electromagnetics*, (Hong Kong, China), City University of Hong Kong, Feb. 2015.
- [12] A. Melendo, A. Coll, M. Pasenau, E. Escolano, and A. Monros, “www.gidhome.com,” 2015. [Online; accessed Nov-2015].
- [13] <http://www.tcl.tk/>. Tcl Developer Xchange.
- [14] J. Ousterhout, *Tcl and the Tk Toolkit*. Addison-Wesley.
- [15] L. E. García-Castillo and M. Salazar-Palma, “Second-order Nédélec tetrahedral element for computational electromagnetics,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (John Wiley & Sons, Inc.)*, vol. 13, pp. 261–287, March-June 2000.
- [16] L. E. Garcia-Castillo, A. J. Ruiz-Genoves, I. Gomez-Revuelto, M. Salazar-Palma, and T. K. Sarkar, “Third-order Nédélec curl-conforming finite element,” *IEEE Transactions on Magnetics*, vol. 38, pp. 2370–2372, Sept. 2002.
- [17] L. A. Piegl and W. Tiller, *The NURBS Book*. Monographs in Visual Communication, Springer, 2nd ed., 1997.
- [18] G. Dantzig and P. Wolfe, “Decomposition principle for linear programs,” *Operations Research*, vol. 8, pp. 101–111, 1960.
- [19] M. Mongiardo and R. Sorrentino, “Efficient and versatile analysis of microwave structures by combined mode matching and finite difference methods,” *IEEE Microwave and Guided Wave Letters*, vol. 3, pp. 241–243, Aug. 1993.
- [20] S. Llorente-Romano, B. Gimeno, and M. Salazar-Palma, “Analysis of cylindrical geometries in rectangular waveguides using mode matching,” in *35th European Microwave Conference*, vol. 2, pp. 4 pp.–, 2005.
- [21] A. Woo, H. Wang, M. Schuh, and M. Sanders, “Em programmer’s notebook-benchmark radar targets for the validation of computational electromagnetics programs,” *Antennas and Propagation Magazine, IEEE*, vol. 35, pp. 84–89, 1993.